به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

درس شبکه‌های عصبی و یادگیری عمیق

تمرین سوم

نام و نام خانوادگی	محمد امانلو – محمد مهدی کعبی
شماره دانشجویی	۸۱۰۱۰۰۰۸۴ - ۸۱۰۱۰۲۵۶۱
تاریخ ارسال گزارش	۱۴۰۱،۰۷.۱۶

فهرست

شکل‌ها

جدول‌ها

No table of figures entries found.

پاسخ ۱. سگمنتیشن تومور مغزی از روی تصاویر MRI

۱,۱- توصیف مدل ارائه شده

نحوه عملکرد مدل پیشنهاد شده:

مدل پیشنهادی در این مقاله ترکیبی از معماری‌های U-Net و VGG16 است که به منظور بهبود عملکرد در مسئله سگمنتیشن تصاویر MRI تومور مغزی طراحی شده است. این مدل با هدف استفاده از قابلیت‌های بالای هر دو معماری، به همراه کاهش پیچیدگی و زمان محاسباتی، به کار گرفته شده است. در حقیقت، این ترکیب سعی دارد از قدرت استخراج ویژگی‌های VGG16 استفاده کند و در عین حال توانایی بازسازی دقیق تصویر را که U-Net ارائه می‌دهد، نیز حفظ کند.

U-Net به دلیل ساختارencoder-decoder توانایی بسیار خوبی در سگمنتیشن تصاویر بیومدیکال دارد. این مدل از طریق مسیر انقباضی ویژگی‌های تصویر را در سطح‌های مختلف استخراج می‌کند و سپس با استفاده از مسیر گسترشی این ویژگی‌ها را به تصویر اصلی بازمی‌گرداند. یکی از ویژگی‌های کلیدی U-Net استفاده از skip connectionها است که به مدل کمک می‌کند تا اطلاعات دقیق از ویژگی‌های مکان‌یابی را به لایه‌های گسترشی منتقل کند و در نتیجه نتیجه سگمنتیشن بهتری ارائه دهد.

اما یکی از چالش‌های U-Net تعداد بالای پارامترها و حجم بالای محاسبات است که به‌خصوص برای پردازش‌های پیچیده مانند تصاویر پزشکی زمان زیادی می‌برد. در مدل پیشنهادی، از VGG16 به عنوان یک مسیر انقباضی جایگزین استفاده شده است. VGG16 یک شبکه کانولوشنی عمیق است که پیشتر روی دیتاست ImageNet آموزش دیده است و توانایی خوبی در استخراج ویژگی‌های پیچیده از تصاویر دارد. در این ترکیب، بخش انقباضی U-Net با VGG16 جایگزین شده است و لایه‌های پایانی VGG16 به منظور بهبود و تطبیق با داده‌های پزشکی فریز نشده‌اند، به این معنی که آن‌ها قابلیت به‌روزرسانی و یادگیری در طول آموزش را دارند.

با استفاده از VGG16 به عنوان بخشی از معماری، تعداد پارامترهای قابل آموزش مدل کاهش یافته و در نتیجه زمان محاسباتی مورد نیاز برای آموزش مدل کاهش یافته است. به علاوه، استفاده از وزن‌های از پیش آموزش دیده VGG16 به بهبود کارایی مدل کمک کرده است، زیرا مدل از ویژگی‌های عمومی که قبلاً روی دیتاست بزرگی یاد گرفته شده است، بهره‌مند می‌شود.

در مسیر انکودر، مدل از لایه‌های کانولوشنی و آپ‌سمپلینگ استفاده می‌کند تا ویژگی‌های استخراج‌شده را به وضوح تصویر اصلی بازسازی کند. همچنین از لایه‌های Batch Normalization و Dropout برای بهبود

پایداری و جلوگیری از overfitting استفاده شده است. این لایه‌ها به مدل کمک می‌کنند تا یادگیری بهتری داشته باشد و عملکرد بهتری بر روی داده‌های اعتبارسنجی و تست نشان دهد.

به منظور آموزش مدل، از ترکیب تابع هزینه Dice Loss و Binary Crossentropy Loss استفاده شده است. Dice Loss به‌خصوص برای مسائل سگمنتیشن مناسب است زیرا بر اساس میزان همپوشانی بین ناحیه‌های پیش‌بینی شده و ناحیه‌های واقعی محاسبه می‌شود. ترکیب این دو تابع هزینه به مدل کمک می‌کند تا عملکرد بهتری در تشخیص ناحیه‌های تومور داشته باشد.

در نهایت، نتایج به دست آمده نشان داد که مدل پیشنهادی دقت بالایی در سگمنتیشن تصاویر MRI تومور مغزی دارد. این مدل توانست با کاهش تعداد پارامترها و استفاده از ویژگی‌های از پیش آموزش دیده، عملکرد بهتری نسبت به نسخه‌های اصلی U-Net ارائه دهد و ناحیه‌های تومور را با دقت خوبی شناسایی کند. استفاده از Transfer Learning و ترکیب معماری‌های مختلف، توانسته است تا مدل با سرعت بیشتر و دقت بالاتری آموزش داده شود و به نتایج بهتری در مقایسه با روش‌های سنتی دست یابد.

ساختار معماری استفاده شده:

مدل معماری پیشنهاد شده در این مقاله ترکیبی است که از ویژگی‌های ساختاری VGG16 و U-Net بهره می‌برد تا مسئله سگمنتیشن تومورهای مغزی در تصاویر MRI را به شکل بهینه حل کند. این ترکیب در حقیقت از بخش‌های اصلی هر دو معماری استفاده می‌کند تا قدرت استخراج ویژگی‌های عمیق VGG16 و قابلیت بازسازی دقیق U-Net را در کنار هم داشته باشد.

مدل از VGG16 به عنوان مسیر انقباضی استفاده می‌کند. VGG16 یکی از مدل‌های شبکه عصبی کانولوشنی معروف است که برای طبقه‌بندی تصاویر طراحی شده و پیشتر روی دیتاست ImageNet آموزش دیده است. معماری VGG16 به دلیل داشتن لایه‌های عمیق و منظم و همچنین تعداد پارامترهای مناسب، در استخراج ویژگی‌های تصویری پیچیده بسیار موفق است. در این مدل ترکیبی، از بخش‌های اولیه VGG16 برای استخراج ویژگی‌ها از تصاویر MRI استفاده شده است. به علاوه، وزن‌های از پیش آموزش دیده VGG16 به عنوان مبنای یادگیری استفاده می‌شوند که باعث کاهش زمان آموزش و بهبود عملکرد مدل در استخراج ویژگی‌های مفید می‌شود.

در مدل ترکیبی، وزن‌های لایه‌های اولیه VGG16 تا حد زیادی فریز شده‌اند، به این معنا که این لایه‌ها در طول فرآیند آموزش تغییر نمی‌کنند. این کار باعث می‌شود که وزن‌های اولیه که از قبل روی دیتاست بزرگ ImageNet یادگیری شده‌اند، برای استخراج ویژگی‌های ابتدایی تصویر استفاده شوند. تنها لایه‌های پایانی VGG16 به منظور بهبود یادگیری باز هستند تا برای تطابق با داده‌های MRI بهینه‌سازی شوند.

بخش گسترشی یا decoder مدل U-Net نیز به عنوان بخش دیگر این معماری استفاده می‌شود. در این بخش، هدف این است که ویژگی‌های استخراج شده در مسیر انقباضی به شکل دقیق بازسازی و تصویر اصلی بازآفرینی شود. این بخش شامل لایه‌های آپ‌سمپلینگ و کانولوشن است که به تدریج ابعاد ویژگی‌ها را به اندازه تصویر اولیه بازمی‌گردانند. همچنین، از skip connectionها استفاده می‌شود تا اطلاعات محلی بهتری از لایه‌های انقباضی به لایه‌های گسترشی منتقل شود و مدل بتواند در بازسازی تصویر نواحی دقیق‌تری را در نظر بگیرد.

یکی از نوآوری‌های کلیدی این معماری ترکیبی استفاده از Dropout و Batch Normalization در مسیر گسترشی است. Dropout به عنوان یک لایه جلوگیری از overfittingعمل می‌کند و با حذف تصادفی برخی از واحدهای نورونی در طول آموزش، کمک می‌کند تا مدل به طور عمومی‌تری یادگیری کند. Batch Normalization نیز به پایداری یادگیری و سرعت همگرایی مدل کمک می‌کند و از تغییرات شدید در توزیع ورودی لایه‌ها جلوگیری می‌کند.

در نهایت، مدل از یک لایه کانولوشنی x۱۱ با فعال‌سازی سیگموید برای تولید خروجی نهایی استفاده می‌کند که نشان‌دهنده ناحیه‌های تومور در تصویر MRI است. این خروجی به شکل یک نقشه باینری است که هر پیکسل آن احتمال تعلق به ناحیه تومور را نشان می‌دهد.

برای آموزش مدل، از ترکیب Binary Crossentropy و Dice Loss به عنوان تابع هزینه استفاده شده است. این ترکیب به مدل کمک می‌کند تا نواحی تومور را به خوبی تشخیص دهد و بخش‌بندی دقیق‌تری انجام دهد. Dice Loss به طور خاص در مسائل سگمنتیشن مفید است زیرا به میزان همپوشانی بین ناحیه‌های پیش‌بینی‌شده و ناحیه‌های واقعی اهمیت می‌دهد.

این معماری ترکیبی در نهایت به گونه‌ای طراحی شده است که از ویژگی‌های عمیق استخراج شده توسط VGG16 و قابلیت‌های بازسازی دقیق U-Net استفاده کند تا دقت و کارایی بالایی در بخش‌بندی تومورهای مغزی از تصاویر MRI ارائه دهد. این مدل توانسته است با ترکیب بهترین ویژگی‌های هر دو معماری، چالش‌های مربوط به بخش‌بندی تصاویر پیچیده پزشکی را به خوبی مدیریت کند و عملکرد بهتری نسبت به روش‌های قبلی ارائه دهد.

نقش هر قسمت در فرآیند سگمنتیشن:

در مرحله اول، مدل از VGG16 به عنوان انکودر استفاده می‌کند. انکودر در حقیقت مسئولیت استخراج ویژگی‌های تصویر را بر عهده دارد. VGG16 به دلیل اینکه یک مدل پیش‌آموزش‌داده‌شده روی مجموعه داده‌های عمومی مانند ImageNet است، به خوبی قادر به استخراج ویژگی‌های پیچیده و عمومی از تصاویر

است. این بخش انکودر شامل چندین لایه کانولوشنی است که تصویر را در چندین مرحله فیلتر می‌کند تا ویژگی‌های مختلفی مانند لبه‌ها، بافت‌ها و اشکال شناسایی شوند. وزن‌های انکودر VGG16 در بیشتر لایه‌ها فریز شده‌اند تا به عنوان ویژگی‌های ثابت مورد استفاده قرار گیرند و تنها لایه‌های انتهایی برای انطباق با داده‌های MRI قابل یادگیری باقی می‌مانند.

پس از انکودر، نوبت به دیکودر می‌رسد که مسئول بازسازی تصویر از ویژگی‌های استخراج‌شده توسط انکودر است. در معماری U-Net، دیکودر با استفاده از عملیات آپ‌سمپلینگ و لایه‌های کانولوشنی به تدریج ویژگی‌های فشرده‌شده را به اندازه و وضوح تصویر اولیه بازمی‌گرداند. این بخش به منظور تولید سگمنت‌های دقیق از تصویر استفاده می‌شود که نشان‌دهنده ناحیه‌های تومور هستند. همچنین در این مرحله از skip connectionها استفاده می‌شود که اتصالات مستقیمی بین لایه‌های انکودر و دیکودر ایجاد می‌کنند. این اتصال‌ها به حفظ اطلاعات موقعیت‌مکانی کمک می‌کنند و مدل را قادر می‌سازند که جزئیات دقیق‌تری را در بازسازی تصویر در نظر بگیرد.

در مسیر دیکودر، لایه‌های Batch Normalization و Dropout به منظور بهبود عملکرد اضافه شده‌اند. لایه Batch Normalization با نرمال‌سازی ورودی‌های لایه‌های بعدی باعث تثبیت و تسریع فرآیند یادگیری می‌شود. این لایه تغییرات زیاد در ورودی‌ها را کاهش می‌دهد و باعث می‌شود که مدل سریع‌تر به همگرایی برسد. از سوی دیگر، لایه Dropout با حذف تصادفی برخی از نورون‌ها در طول آموزش، به جلوگیری از overfittingکمک می‌کند. این کار باعث می‌شود که مدل به صورت کلی‌تری یاد بگیرد و وابستگی زیادی به نورون‌های خاص نداشته باشد.

در نهایت، مدل با یک لایه کانولوشنی x11 همراه با تابع فعال‌سازی سیگموید خروجی نهایی را تولید می‌کند. این لایه خروجی باینری تولید می‌کند که نشان می‌دهد هر پیکسل به ناحیه تومور تعلق دارد یا خیر. این خروجی به شکل یک نقشه احتمال باینری است که برای تعیین نواحی تومور مغزی استفاده می‌شود. هر پیکسل در این نقشه نمایانگر احتمال تعلق آن به ناحیه تومور است که با استفاده از تابع سیگموید مقادیر بین ۰ و ۱ می‌گیرد.

برای آموزش مدل، از ترکیب Binary Crossentropy و Dice Loss به عنوان تابع هزینه استفاده شده است. Binary Crossentropy تابعی مناسب برای مشکلات باینری است و با استفاده از تفاوت‌های بین مقادیر پیش‌بینی‌شده و مقادیر واقعی، خطا را محاسبه می‌کند. Dice Loss به طور ویژه در مسائل سگمنتیشن مورد استفاده قرار می‌گیرد و به میزان همپوشانی بین نواحی پیش‌بینی‌شده و نواحی واقعی تمرکز دارد. ترکیب این دو تابع هزینه به مدل کمک می‌کند تا دقت بیشتری در تشخیص و تفکیک نواحی تومور داشته باشد.

به‌طور کلی، مدل ترکیبی UNet-VGG16 شامل سه بخش اصلی است: انکودر که ویژگی‌ها را استخراج می‌کند، دیکودر به ویژگی‌ها را به وضوح تصویر اصلی بازسازی می‌کند، و خروجی که نواحی تومور را به صورت باینری نمایش می‌دهد. هر بخش با ایفای نقش خاص خود به بهبود عملکرد نهایی در مسئله سگمنتیشن تصاویر MRI تومور مغزی کمک می‌کند. این مدل با ترکیب قدرت استخراج ویژگی‌های عمومی VGG16 و قابلیت بازسازی دقیق U-Net توانسته است عملکرد بهتری نسبت به مدل‌های معمولی ارائه دهد و دقت بالایی در شناسایی تومورها داشته باشد.

توضیح دهید که چرا معماری VGGبرای یادگیری انتقالی انتخاب شده است.

اولین دلیل انتخاب VGG16، عمق و ساختار متقارن این مدل است. VGG16 یکی از مدل‌های موفق در شبکه‌های عصبی کانولوشنی است که با عمق ۱۶ لایه و ساختار متقارن و ساده، به خوبی می‌تواند ویژگی‌های پیچیده را از تصاویر استخراج کند. هر لایه در این معماری به خوبی ویژگی‌های متفاوتی از تصویر را شناسایی می‌کند؛ لایه‌های ابتدایی ویژگی‌های عمومی مانند لبه‌ها و بافت‌ها و لایه‌های انتهایی ویژگی‌های پیچیده‌تر مانند اشکال و الگوها را شناسایی می‌کنند. این ساختار متقارن و پیشرفته باعث می‌شود که VGG16 در بسیاری از مسائل تصویری عملکرد فوق‌العاده‌ای داشته باشد.

یکی دیگر از دلایل انتخاب VGG16، وزن‌های از پیش آموزش دیده آن است. VGG16 پیشتر روی دیتاست ImageNet که شامل میلیون‌ها تصویر و هزاران دسته‌بندی مختلف است، آموزش دیده است. این به این معناست که وزن‌های VGG16 به‌خوبی قادر به استخراج ویژگی‌های عمومی از تصاویر هستند. وقتی ما از این وزن‌های از پیش آموزش دیده در مسئله‌ی جدیدی مانند سگمنتیشن تصاویر MRI استفاده می‌کنیم، نیاز نیست که مدل را از صفر آموزش دهیم و می‌توانیم از ویژگی‌های عمومی استخراج‌شده توسط لایه‌های ابتدایی VGG16 بهره ببریم. این امر باعث می‌شود که زمان محاسباتی برای آموزش مدل کاهش یابد و همچنین نیاز به داده‌های زیاد برای آموزش کمتر شود.

دلیل سوم استفاده از VGG16، قابلیت تعمیم بهتر است. زمانی که از یک مدل از پیش آموزش‌دیده استفاده می‌کنیم، این مدل قبلاً با داده‌های متنوعی آموزش دیده و توانایی شناسایی ویژگی‌های مختلف را دارد. این ویژگی‌ها می‌توانند به خوبی در مسائل جدید تعمیم پیدا کنند و باعث شوند که مدل جدید نیز با داده‌های مشابه و البته متفاوت، عملکرد خوبی داشته باشد. بنابراین، استفاده از VGG16 برای استخراج ویژگی‌های تصویری به‌خصوص در مشکلاتی که داده‌های کمتری در اختیار داریم، مفید است و منجر به کاهش خطای یادگیری می‌شود.

همچنین، انتخاب VGG16 به دلیل سادگی و کارایی آن در مقایسه با مدل‌های پیچیده‌تر مانند ResNet یا Inception است. VGG16 با اینکه عمق نسبتاً زیادی دارد، اما ساختار آن بسیار ساده است و لایه‌های آن به صورت پشت‌سرهم قرار گرفته‌اند. این سادگی باعث می‌شود که مدل به راحتی قابل پیاده‌سازی و بهینه‌سازی باشد و در مسائل مختلف استفاده شود. از آنجایی که مسئله‌ی سگمنتیشن تصاویر MRI به دقت بالایی نیاز دارد، استفاده از معماری‌های پیچیده‌تر ممکن است باعث افزایش خطر overfitting شود. در مقابل، VGG16 با تعادلی که بین عمق و سادگی برقرار کرده است، به راحتی قابل بهینه‌سازی است و می‌تواند دقت مناسبی ارائه دهد.

ساختار لایه‌ای VGG16 نیز به نوعی است که به راحتی با معماری‌های دیگر مانند U-Net ترکیب می‌شود. لایه‌های VGG16 در واقع به عنوان انکودر استفاده می‌شوند که ویژگی‌های تصویر را استخراج می‌کنند و سپس این ویژگی‌ها به بخش دیکودر U-Net ارسال می‌شوند تا سگمنتیشن انجام شود. این ترکیب از نظر معماری بسیار هماهنگ است و باعث می‌شود که ویژگی‌های سطح بالا که توسط VGG16 استخراج شده‌اند به خوبی با فرآیند بازسازی تصویر توسط U-Net همخوانی داشته باشند.

یادگیری انتقالی در این مدل از چه جهات میتواند کمک کننده باشد

کاهش زمان آموزش: یکی از اصلی‌ترین مزایای استفاده از Transfer Learning در این مدل، کاهش زمان مورد نیاز برای آموزش مدل است. در مدل پیشنهادی، بخش انکودر از معماری VGG16 استفاده می‌کند که پیش‌تر روی دیتاست ImageNet آموزش دیده است. این یعنی وزن‌های این مدل برای استخراج ویژگی‌های عمومی از تصاویر، از قبل بهینه شده‌اند. به همین دلیل، نیازی به آموزش مدل از صفر نیست و بسیاری از مراحل ابتدایی استخراج ویژگی‌ها (مانند شناسایی لبه‌ها، بافت‌ها و اشکال ساده) به سرعت انجام می‌شود. این امر به خصوص در مشکلات پیچیده مانند تصاویر پزشکی که داده‌های زیادی نداریم، می‌تواند بسیار کمک‌کننده باشد.

استفاده از ویژگی‌های از پیش یادگرفته‌شده: مدل VGG16 پیش‌تر روی دیتاست بسیار بزرگی که شامل میلیون‌ها تصویر است، آموزش دیده است و ویژگی‌های مختلفی را از تصاویر یاد گرفته است. این ویژگی‌ها از ویژگی‌های ساده مانند لبه‌ها و بافت‌ها شروع می‌شوند و به ویژگی‌های پیچیده‌تر مانند اشکال و ساختارهای کلی می‌رسند. استفاده از این وزن‌ها در مدل UNet-VGG16 باعث می‌شود که مدل بتواند از ویژگی‌های از پیش یادگرفته‌شده برای استخراج اطلاعات مفید از تصاویر MRI استفاده کند. این ویژگی‌ها برای تشخیص ساختارهای موجود در تصاویر پزشکی مانند تومورها و نواحی مختلف مغز بسیار کمک‌کننده هستند.

کاهش نیاز به داده‌های آموزشی بزرگ: یکی از چالش‌های اساسی در آموزش مدل‌های عمیق، نیاز به داده‌های آموزشی زیاد است. به‌خصوص در حوزه‌های پزشکی، جمع‌آوری داده‌های بزرگ و متنوع بسیار دشوار و هزینه‌بر است. استفاده از Transfer Learning به این معناست که مدل از ویژگی‌های از پیش یادگرفته‌شده روی یک دیتاست بزرگ بهره‌برداری می‌کند، و به این ترتیب نیاز به تعداد زیادی داده‌های آموزشی کاهش پیدا می‌کند. مدل می‌تواند با تعداد کمتری داده‌های پزشکی، عملکرد مناسبی داشته باشد و به دقت بالایی برسد.

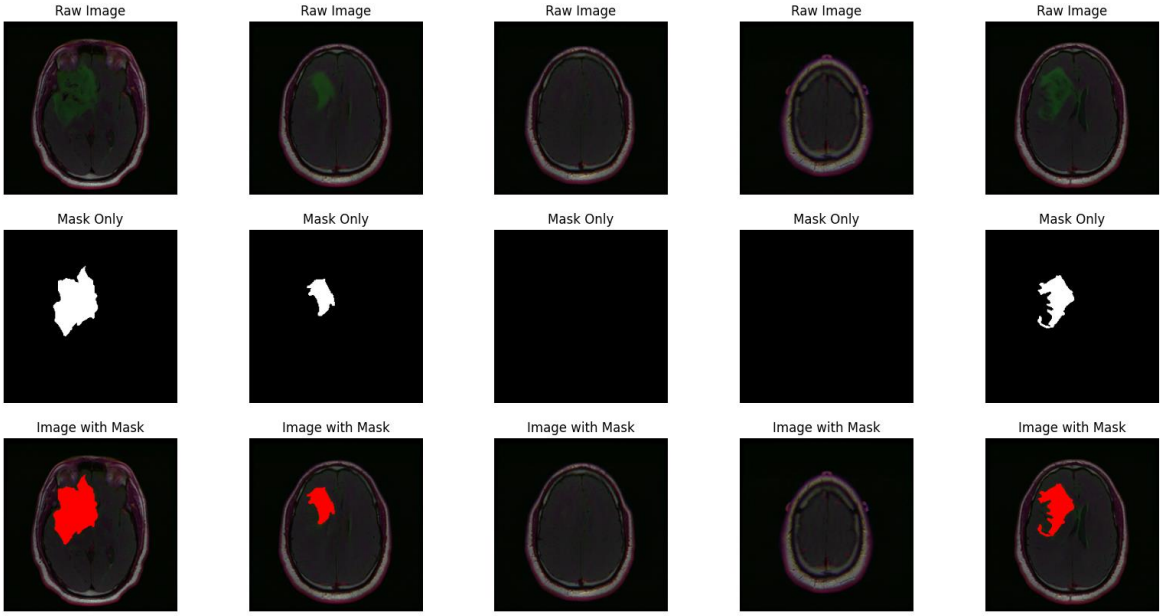
کاهش خطرoverfiting یکی دیگر از مزایای استفاده از Transfer Learning در این مدل، کاهش خطر بیش‌برازش است. در مسائل پزشکی که معمولاً داده‌های آموزشی محدودی در دسترس است، آموزش مدل‌های پیچیده از صفر می‌تواند منجر به بیش‌برازش شود، یعنی مدل به جای یادگیری ویژگی‌های عمومی، به داده‌های خاص آموزشی وابسته می‌شود. استفاده از وزن‌های از پیش یادگرفته‌شده در VGG16 به مدل کمک می‌کند تا ویژگی‌های عمومی را بهتر یاد بگیرد و از وابستگی بیش از حد به داده‌های خاص جلوگیری کند. این کار به بهبود توانایی مدل در تعمیم دادن و عملکرد بهتر روی داده‌های جدید و دیده‌نشده کمک می‌کند.

شروع آموزش از نقطه‌ای بهینه‌تر: در روش Transfer Learning، به جای اینکه مدل آموزش خود را از وزن‌های تصادفی آغاز کند، از وزن‌های بهینه‌ای که از قبل به دست آمده‌اند، استفاده می‌کند. این امر باعث می‌شود که مدل از نقطه‌ای بهینه‌تر شروع به یادگیری کند و نیاز به تنظیمات طولانی مدت نداشته باشد. این موضوع باعث می‌شود که سرعت همگرایی مدل افزایش یابد و مدل سریع‌تر به نتایج مطلوب برسد.

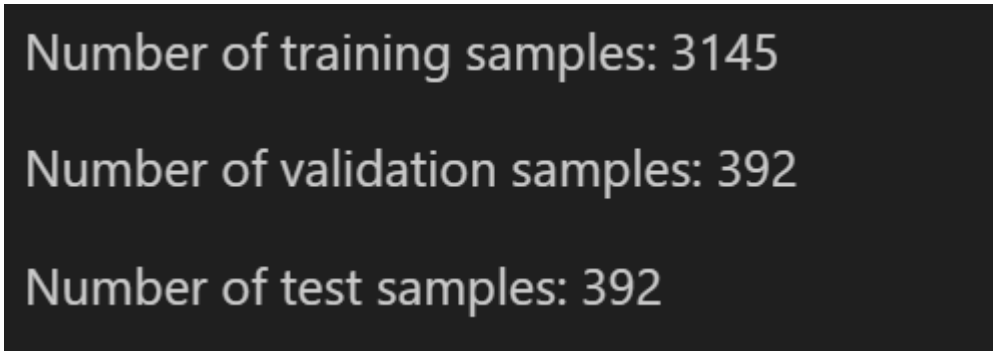
بهبود دقت مدل در مسائل پیچیده: در مسئله‌ی سگمنتیشن تومورهای مغزی، نیاز به دقت بالا و تشخیص دقیق نواحی تومور وجود دارد. استفاده از Transfer Learning باعث می‌شود که مدل با دقت بالاتری ویژگی‌های مربوط به ساختارهای تومور را از ویژگی‌های دیگر مغز جدا کند. به دلیل اینکه مدل VGG16 قبلاً روی دیتاست بزرگ و متنوعی آموزش دیده است، می‌تواند به خوبی ویژگی‌های پیچیده و منحصر به فرد را شناسایی کند و از این دانش برای تشخیص تومورها در تصاویر پزشکی استفاده کند.

کاهش پیچیدگی و نیاز به تنظیم‌های دقیق: در بسیاری از موارد، آموزش مدل‌های عمیق نیازمند تنظیم دقیق و آزمون و خطا در مورد پارامترهای مختلف مانند نرخ یادگیری، تعداد لایه‌ها و batch size است. استفاده از Transfer Learning این پیچیدگی‌ها را کاهش می‌دهد، زیرا بخش بزرگی از این پارامترها در مراحل پیشین آموزش مدل تنظیم شده‌اند و نیاز به تغییر زیادی ندارند. این امر به سادگی فرآیند آموزش کمک می‌کند و نیاز به تلاش‌های مکرر برای پیدا کردن بهترین تنظیمات را کاهش می‌دهد.

۱,۲-آماده سازی مجموعه داده



شکل ۱نمونه هایی از داده های اصلی



شکل ۲تعداد داده های هر دسته

```python
def load_and_split_data(img_path):
    if not os.path.exists('TRAIN'):
        os.makedirs('TRAIN')
    if not os.path.exists('VAL'):
        os.makedirs('VAL')
    if not os.path.exists('TEST'):
        os.makedirs('TEST')

    for class_folder in os.listdir(img_path):
        class_path = os.path.join(img_path, class_folder)
        if not os.path.isdir(class_path):
            continue
        img_num = len(os.listdir(class_path))

        for n, file_name in enumerate(os.listdir(class_path)):
            if file_name.endswith(".tif") and not file_name.endswith("_mask.tif"):
                img = os.path.join(class_path, file_name)
                mask = img.replace(".tif", "_mask.tif")

                if not os.path.exists(mask):
                    print(f"Warning: Mask file not found for image {img}")
                    continue

                if n < 5:
                    dest_path = os.path.join('TEST', class_folder.upper())
                elif n < 0.8 * img_num:
                    dest_path = os.path.join('TRAIN', class_folder.upper())
                else:
                    dest_path = os.path.join('VAL', class_folder.upper())

                if not os.path.exists(dest_path):
                    os.makedirs(dest_path)

                shutil.copy(img, os.path.join(dest_path, file_name))
                shutil.copy(mask, os.path.join(dest_path, os.path.basename(mask)))

load_and_split_data(os.path.join(path, 'kaggle_3m'))
```

شکل ۳تقسیم دادگان به ۳ دسته آموزش، اعتبارسنجی و ارزیابی

ما در این پروژه ابتدا مجموعه داده مربوط به تصاویر MRI مغز که شامل تصاویر اصلی و ماسک‌های مربوط به نواحی تومور است را از منبع مورد نظر دانلود کردیم. این مجموعه داده شامل تصاویر مغز بیماران مختلف است که به صورت TIFF ذخیره شده‌اند و ماسک‌ها نیز به صورت فایل‌های جداگانه با پسوند mask.tif_ موجود هستند. هدف ما این بود که این تصاویر را برای مسئله سگمنتیشن تومور مغزی پردازش کنیم.

در اولین گام، ما تصاویر و ماسک‌های مربوطه را از پوشه اصلی خواندیم و آن‌ها را در سه بخش به نسبت ۱۰-۱۰-۸۰ به مجموعه داده‌های آموزش، اعتبارسنجی و تست تقسیم کردیم. این تقسیم‌بندی به گونه‌ای انجام شد که ۸۰ درصد داده‌ها برای آموزش مدل، ۱۰ درصد برای اعتبارسنجی مدل در حین آموزش و ۱۰ درصد باقی‌مانده برای ارزیابی نهایی عملکرد مدل اختصاص داده شوند. این کار به ما کمک می‌کند تا مدل را به شکلی بهتر آموزش دهیم و از بروز مشکلاتی مانند بیش‌برازش جلوگیری کنیم.

برای هر تصویر، ماسک مربوطه را نیز بررسی کردیم و اطمینان حاصل کردیم که هر جفت تصویر و ماسک در یک دسته قرار گیرند. اگر تصویری بدون ماسک یافت می‌شد، آن تصویر از مجموعه داده حذف می‌گردید. در ادامه، داده‌ها به سایز ۲۵۶ * ۲۵۶ تغییر اندازه داده شدند تا بتوانیم آن‌ها را در مدل مورد استفاده قرار دهیم. سپس، تصاویر به مقیاس ۰ تا ۱ نرمال‌سازی شدند تا مدل یادگیری بهتری داشته باشد.

برای مشاهده و بررسی کیفیت داده‌ها، چند نمونه از تصاویر اصلی همراه با ماسک‌های آن‌ها را نمایش دادیم. این نمایش شامل سه حالت بود: تصویر خام، ماسک تنها، و تصویر همراه با ماسک ترکیب‌شده که ناحیه تومور را با رنگ قرمز مشخص می‌کند. این نمایش به ما کمک کرد تا از کیفیت داده‌ها و درستی ماسک‌های موجود اطمینان حاصل کنیم.

پس از آماده‌سازی داده‌ها، تعداد نمونه‌های هر بخش را گزارش کردیم. به طور دقیق، تعداد داده‌های موجود در هر بخش به شرح زیر بود:

تعداد نمونه‌های آموزش: 3145

تعداد نمونه‌های اعتبارسنجی: 392

تعداد نمونه‌های تست: 392

این تقسیم‌بندی تضمین می‌کند که مدل ما به خوبی آموزش دیده و همچنین قابلیت تعمیم مناسبی روی داده‌های جدید دارد. استفاده از مجموعه داده اعتبارسنجی به ما کمک کرد که در حین آموزش، عملکرد مدل را نظارت کنیم و با استفاده از معیارهایی مانند دقت و ضرر، از بروز مشکلات احتمالی جلوگیری کنیم. در نهایت، مجموعه داده تست برای ارزیابی نهایی مدل استفاده خواهد شد تا عملکرد آن در شرایط واقعی ارزیابی شود.

۱,۳-تقویت داده

در این بخش از پروژه، ما بر روی تقویت داده‌ها تمرکز کردیم تا بتوانیم تنوع مجموعه داده‌های خود را افزایش دهیم و به این ترتیب، مدل بهتری برای شناسایی تومورهای مغزی بر روی تصاویر MRI ایجاد کنیم. به‌طور کلی، تقویت داده به ما این امکان را می‌دهد که مجموعه داده‌های کوچک یا با تنوع کم را به مجموعه‌ای بزرگ‌تر و متنوع‌تر تبدیل کنیم تا مدل بتواند توانایی تعمیم‌دهی بهتری در مواجهه با داده‌های جدید پیدا کند. این مسئله به‌ویژه در پروژه‌های پزشکی که جمع‌آوری داده‌های متنوع و با کیفیت

چالش‌برانگیز است، بسیار اهمیت دارد. در ادامه به‌طور کامل و مفصل در مورد فرآیند تقویت داده‌هایی که انجام دادیم و مزایای آن توضیح می‌دهیم.

برای شروع، هدف اصلی ما افزایش تعداد و تنوع نمونه‌ها بود تا مدل بتواند ویژگی‌های مختلف تومور را در شرایط گوناگون یاد بگیرد. برای این منظور، از تکنیک‌های مختلفی شامل چرخش، تغییر مقیاس ، جابجایی افقی و عمودی ، وارونگی افقی ، و تغییرات شدت روشنایی استفاده کردیم. هر یک از این تکنیک‌ها نقش ویژه‌ای در بهبود توانایی مدل در یادگیری ویژگی‌های مهم نواحی تومور داشتند.

ابتدا، ما از تکنیک چرخش تصاویر استفاده کردیم. چرخش تصاویر به ما کمک کرد که تنوع زاویه‌ای تصاویر را افزایش دهیم. در واقع، تصاویر MRI ممکن است به دلایل مختلف مانند موقعیت قرارگیری بیمار یا نحوه تصویربرداری در زوایای مختلف ثبت شوند. اگر مدل فقط بر اساس تصاویر با زاویه خاص آموزش ببیند، ممکن است نتواند تصاویر با زاویه‌های مختلف را به‌خوبی تحلیل کند. بنابراین، با اعمال چرخش تصادفی به تصاویر، مدل مجبور شد که بدون توجه به زاویه تصویر، ویژگی‌های مهم آن را یاد بگیرد. ما زاویه چرخش را در محدوده ۲۰ درجه تنظیم کردیم تا چرخش‌های کوچکی که ممکن است به‌طور طبیعی در تصاویر ایجاد شود، شبیه‌سازی شود. این کار باعث شد که مدل در مواجهه با داده‌های ناآشنا که ممکن است زاویه دید متفاوتی داشته باشند، عملکرد مناسبی داشته باشد.

یکی دیگر از تکنیک‌های مهمی که به کار بردیم، تغییر مقیاس بود. ما با اعمال تغییرات تصادفی در مقیاس تصاویر، به مدل کمک کردیم که بتواند تومورها را در اندازه‌های مختلف شناسایی کند. این مسئله به‌خصوص در شرایطی که تومورها می‌توانند بسیار کوچک یا بزرگ باشند، اهمیت دارد. تغییر مقیاس تصاویر به مدل این امکان را می‌دهد که بدون توجه به اندازه تومور، به دنبال الگوهای کلیدی بگردد. این ویژگی باعث شد که مدل در برابر تغییرات اندازه مقاوم‌تر باشد و بتواند نواحی تومور را با اندازه‌های مختلف شناسایی کند.

جابجایی افقی و عمودی نیز یکی از تکنیک‌های مؤثر بود که به مدل کمک کرد نسبت به تغییرات مکان تومور مقاومت بیشتری داشته باشد. در عمل، موقعیت قرارگیری تومور در مغز می‌تواند متفاوت باشد و همین امر ممکن است منجر به قرارگیری تومور در بخش‌های مختلف تصویر MRI شود. با اعمال جابجایی‌های افقی و عمودی به‌طور تصادفی، مدل یاد گرفت که بدون توجه به موقعیت دقیق تومور، الگوهای مرتبط با آن را شناسایی کند. این کار به افزایش توانایی مدل در شناسایی تومورهایی که در نقاط مختلف مغز ظاهر می‌شوند، کمک شایانی کرد.

برای افزایش تنوع تصاویر، از وارونگی افقی نیز استفاده کردیم. وارونگی افقی به این معنی است که تصویر از چپ به راست وارونه می‌شود. این تکنیک به مدل کمک کرد که ویژگی‌های تصویر را بدون توجه به جهت آن یاد بگیرد. به بیان دیگر، اگر تومور در قسمت چپ تصویر باشد و مدل تنها با داده‌هایی آموزش دیده

باشد که تومور در همین بخش قرار دارد، ممکن است نتواند تومورهایی که در جهت‌های مخالف هستند را به‌خوبی شناسایی کند. بنابراین، با وارونگی افقی، مدل مجبور شد که ویژگی‌های تصویر را فارغ از جهت آن‌ها یاد بگیرد و این باعث افزایش مقاومت مدل در برابر تغییر جهت ویژگی‌های تصویر شد.

یکی دیگر از تکنیک‌هایی که استفاده کردیم، تغییر شدت روشنایی بود. شدت روشنایی تصاویر MRI می‌تواند به دلایل مختلفی مانند تنظیمات دستگاه، شرایط تصویربرداری، یا تفاوت‌های فردی در بیماران متفاوت باشد. با استفاده از تغییرات شدت روشنایی، ما به مدل کمک کردیم که بتواند در شرایط نوری مختلف ویژگی‌های تومور را تشخیص دهد. این تکنیک باعث شد که مدل در مواجهه با تصاویر با شدت نور بالا یا پایین، توانایی شناسایی نواحی تومور را حفظ کند و به‌صورت دقیق به تحلیل تصویر بپردازد. ما تغییرات روشنایی را در محدوده ۰٫۸ تا ۱٫۱ تنظیم کردیم تا تصاویر به‌صورت کمی تاریک‌تر یا روشن‌تر شوند و به این ترتیب مدل در شرایط نوری مختلف عملکرد بهتری داشته باشد.

در طی فرآیند تقویت داده‌ها، ما چندین نمونه از تصاویر آموزشی را انتخاب کردیم و آن‌ها را با استفاده از تکنیک‌های ذکر شده تقویت کردیم. برای هر تصویر و ماسک مربوط به آن، نسخه‌های تقویت‌شده‌ای ایجاد کردیم که شامل تغییرات زاویه، اندازه، مکان، روشنایی، و وارونگی بودند. این تصاویر جدید به مجموعه داده‌های آموزشی اصلی اضافه شدند تا مدل با مجموعه‌ای متنوع‌تر از تصاویر روبه‌رو شود. هدف ما این بود که مدل ویژگی‌های کلی‌تری از تصاویر یاد بگیرد و از یادگیری جزئیات خاص هر تصویر جلوگیری شود، که این امر به جلوگیری از overfittingکمک شایانی کرد.

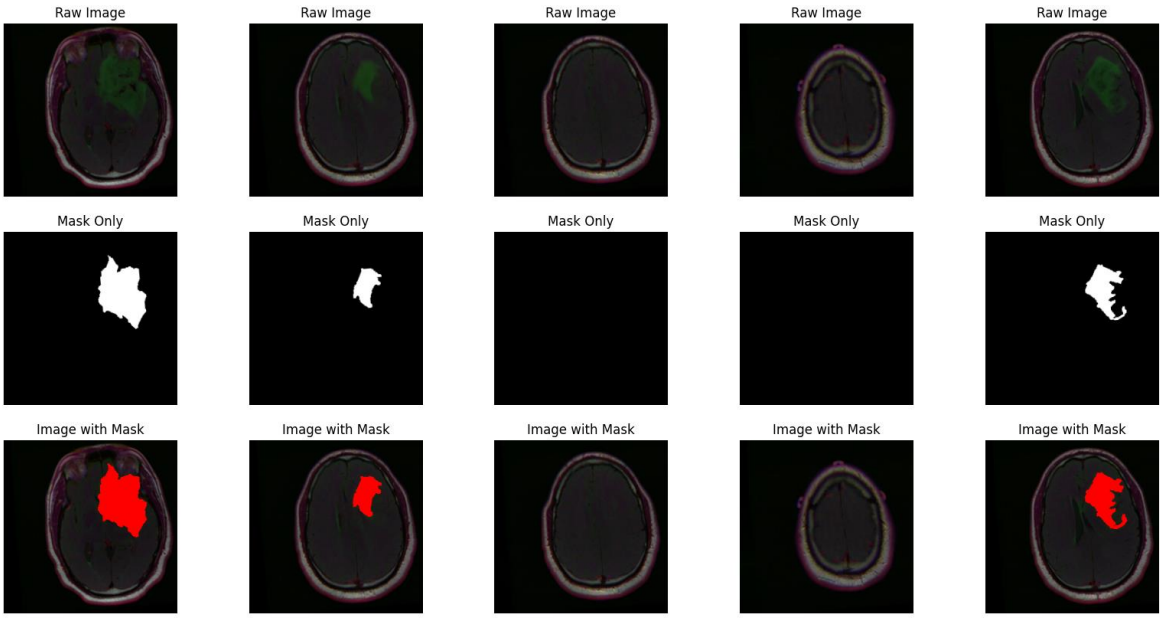
نتایج به‌دست‌آمده از این فرآیند نشان داد که تقویت داده‌ها نقش بسیار مهمی در بهبود عملکرد مدل داشتند. Dice Coefficient که یکی از معیارهای اصلی برای سنجش دقت پیش‌بینی مدل است، پس از اعمال تقویت داده‌ها به‌طور محسوسی افزایش یافت. این معیار نشان می‌دهد که مدل تا چه حد توانسته نواحی پیش‌بینی‌شده و نواحی واقعی را همپوشانی کند. پس از اعمال تقویت داده‌ها، مدل توانست نواحی تومور را با دقت بیشتری شناسایی کند و این موضوع در افزایش Dice Coefficient نمایان شد. علاوه بر آن، IoU Score (Intersection over Union) که شاخص دیگری برای ارزیابی دقت سگمنتیشن است نیز افزایش یافت، که نشان‌دهنده بهبود عملکرد مدل در تشخیص نواحی تومور بود.

تقویت داده‌ها همچنین باعث افزایش توانایی تعمیم‌دهی مدل شد. در واقع، مدل با مشاهده تصاویر متنوع‌تر و گوناگون توانست ویژگی‌های کلی‌تری از داده‌ها یاد بگیرد که به آن کمک کرد تا در مواجهه با داده‌های جدید و ناآشنا نیز عملکرد مناسبی داشته باشد. این امر باعث شد که مدل در مواجهه با داده‌های اعتبارسنجی و تست نیز عملکرد بهتری نشان دهد و دچار کاهش دقت نشود. همچنین، این افزایش

تعمیم‌دهی به مدل کمک کرد که بتواند نواحی تومور را بدون توجه به شرایط نوری، زاویه، یا موقعیت دقیق تومور به‌درستی شناسایی کند.

```
data_gen_args = dict(rotation_range=20,
                     width_shift_range=0.1,
                     height_shift_range=0.1,
                     shear_range=0.1,
                     zoom_range=0.1,
                     horizontal_flip=True,
                     fill_mode='nearest',
                     brightness_range = (0.8,1.1))
```

شکل ۴روش های تقویت داده



شکل ۵نمونه هایی از دادگان تقویت شده

۱,۴- بهینه ساز، معیارها و تابع هزینه

معیار IoU Score چیست؟

معیار IoU Score یا همان Intersection over Union یکی از مهم‌ترین و پرکاربردترین معیارهای ارزیابی در مسائل سگمنتیشن تصویر است، به‌ویژه در زمینه یادگیری عمیق و بینایی ماشین. این معیار برای اندازه‌گیری دقت پیش‌بینی مدل در تشخیص و سگمنت کردن نواحی مشخصی از یک تصویر، مانند تومورهای مغزی یا اشیاء خاص، به کار می‌رود.

برای درک مفهوم IoU بهتر است که ابتدا به تعریف سگمنتیشن بپردازیم. در مسائل سگمنتیشن تصویر، هدف اصلی مدل این است که برای هر پیکسل تصویر، تعیین کند که آیا متعلق به یک شیء خاص (مثلاً تومور) است یا خیر. به عبارت دیگر، در سگمنتیشن، مدل تلاش می‌کند نواحی موردنظر را از سایر بخش‌های تصویر جدا کند.

برای ارزیابی عملکرد مدل در چنین شرایطی، IoU به‌عنوان یک معیار کلیدی مطرح می‌شود. به‌طور کلی، Intersection over Union نشان‌دهنده نسبت بین تقاطع ((intersectionو اتحاد ((unionدو مجموعه است که این دو مجموعه عبارت‌اند از: منطقه پیش‌بینی‌شده توسط مدل و منطقه واقعی که به عنوان Ground Truth مشخص شده است.

از نظر ریاضی، معیار IoU به‌صورت نسبت تعداد پیکسل‌های مشترک بین ناحیه پیش‌بینی‌شده و ناحیه واقعی به تعداد پیکسل‌های موجود در اتحاد این دو ناحیه تعریف می‌شود. به‌عبارتی دیگر، IoU برابر است با نسبت تعداد پیکسل‌هایی که هم در ناحیه پیش‌بینی‌شده و هم در ناحیه واقعی حضور دارند، به تعداد تمامی پیکسل‌هایی که در حداقل یکی از این دو ناحیه وجود دارند. IoU به‌طور کلی یک عدد بین صفر و یک است که نشان‌دهنده میزان تطابق یا همپوشانی بین ناحیه پیش‌بینی‌شده و ناحیه واقعی است. هرچه مقدار IoU به یک نزدیک‌تر باشد، نشان‌دهنده دقت بالاتر پیش‌بینی مدل و تطابق بیشتر بین ناحیه پیش‌بینی‌شده و ناحیه واقعی است.

برای محاسبه IoU در یک تصویر، مراحل زیر را طی می‌کنیم: ابتدا باید دو ماسک در اختیار داشته باشیم. ماسک واقعی که ناحیه موردنظر را نشان می‌دهد، به عنوان Ground Truth در نظر گرفته می‌شود. ماسک پیش‌بینی‌شده، خروجی مدل است که به‌طور اتوماتیک توسط مدل تولید شده است. سپس تقاطع این دو ماسک به معنای تعداد پیکسل‌هایی است که هم در ماسک واقعی و هم در ماسک پیش‌بینی‌شده حضور دارند. به بیان دیگر، این تعداد پیکسل‌ها نشان‌دهنده آن بخش از تصویر است که هم توسط مدل درست شناسایی شده و هم در واقعیت وجود دارد. بعد از آن، اتحاد این دو ماسک، شامل تمامی پیکسل‌هایی است که در حداقل یکی از ماسک‌های واقعی یا پیش‌بینی‌شده وجود دارند. در نهایت، IoU برابر است با نسبت تعداد پیکسل‌های تقاطع به تعداد پیکسل‌های اتحاد.

IoU یکی از معیارهای اصلی برای سنجش دقت مدل در مسائل سگمنتیشن است، اما معیارهای دیگری نیز برای این منظور وجود دارند، از جمله Dice Coefficient. Dice Coefficient نیز معیاری برای اندازه‌گیری همپوشانی بین دو مجموعه است، اما به نوعی حساسیت بیشتری به تعداد پیکسل‌های مشترک دارد. می‌توان نشان داد که ارتباط بین IoU و Dice به این شکل است که Dice از IoU مشتق می‌شود و هر دو معیار اطلاعات مشابهی در مورد میزان همپوشانی ارائه می‌دهند، اما IoU به عنوان یک معیار جامع‌تر و پرکاربردتر در ارزیابی مسائل سگمنتیشن مطرح است.

در یادگیری عمیق، به‌ویژه در شبکه‌های عصبی کانولوشنی رای مسائل سگمنتیشن، IoU به عنوان یکی از معیارهای اصلی برای ارزیابی عملکرد مدل و همچنین بهینه‌سازی آن در فرآیند آموزش به کار می‌رود. در بسیاری از موارد، از IoU به عنوان تابع هزینه نیز استفاده می‌شود، به‌طوری که مدل تلاش می‌کند با کمینه‌سازی اختلاف بین ناحیه پیش‌بینی‌شده و ناحیه واقعی، مقدار IoU را به حداکثر برساند.

از مزایای IoU می‌توان به سادگی و شهودی بودن آن اشاره کرد. IoU یک معیار ساده و قابل فهم است که به‌راحتی می‌توان آن را با مفهوم همپوشانی نواحی توضیح داد. همچنین، این معیار قابلیت تعمیم‌دهی به مسائل مختلف دارد و می‌تواند در انواع مسائل سگمنتیشن و تشخیص شیء به کار رود و به عنوان یک معیار استاندارد برای ارزیابی عملکرد مدل‌ها در بسیاری از پروژه‌های یادگیری عمیق استفاده شود. با این حال، یکی از معایب IoU این است که اگر ناحیه تومور بسیار کوچک باشد و مدل نتواند آن را به‌درستی شناسایی کند، مقدار IoU به شدت کاهش می‌یابد و به صفر نزدیک می‌شود. این مسئله می‌تواند باعث شود که مدل عملکرد ضعیفی روی نواحی کوچک نشان دهد. همچنین، IoU تنها به میزان همپوشانی بین دو ناحیه اهمیت می‌دهد و به موقعیت دقیق آن‌ها توجهی ندارد. به همین دلیل، ممکن است در برخی موارد، ناحیه پیش‌بینی‌شده در موقعیت نادرستی قرار داشته باشد ولی به دلیل همپوشانی زیاد، مقدار IoU بالا باشد. در فرآیند آموزش مدل، می‌توان از IoU به عنوان یک معیار برای مقایسه مدل‌های مختلف یا ارزیابی دقت در طول آموزش استفاده کرد. همچنین، از IoU می‌توان به عنوان یک معیار برای انتخاب بهترین مدل در طول آموزش و اعتبارسنجی استفاده کرد. برای مثال، مدلی که بیشترین مقدار IoU را در داده‌های اعتبارسنجی داشته باشد، به عنوان مدل نهایی انتخاب می‌شود.

Dice Coefficient را توضیح دهید

Dice Coefficient یکی از معیارهای اصلی و پرکاربرد برای سنجش دقت مدل‌های سگمنتیشن تصویر است. این معیار به‌ویژه در مسائل مربوط به سگمنتیشن تصویر و یادگیری عمیق بسیار مورد استفاده قرار می‌گیرد و برای اندازه‌گیری میزان همپوشانی بین ناحیه پیش‌بینی‌شده و ناحیه واقعی به کار می‌رود. Dice Coefficient ارتباط نزدیکی با معیار IoU (Intersection over Union) دارد و معمولاً در کنار آن استفاده

می‌شود. از نظر مفهومی، Dice Coefficient به طور خاص برای اندازه‌گیری میزان همپوشانی بین دو مجموعه طراحی شده است. در مسائل سگمنتیشن تصویر، این دو مجموعه عبارت‌اند از: منطقه واقعی یا همون Ground Truthکه توسط متخصص یا به‌طور دستی مشخص شده و منطقه پیش‌بینی‌شده توسط مدل که به صورت خودکار توسط مدل یادگیری عمیق تولید شده است. این معیار به‌عنوان یک شاخص برای ارزیابی دقت مدل به کار می‌رود و نشان می‌دهد که چقدر ناحیه پیش‌بینی‌شده توسط مدل با ناحیه واقعی همپوشانی دارد.

از نظر ریاضی، Dice Coefficient به صورت زیر تعریف می‌شود:

Dice برابر است با دو برابر تعداد پیکسل‌های مشترک بین ناحیه پیش‌بینی‌شده و ناحیه واقعی، تقسیم بر مجموع تعداد کل پیکسل‌ها در هر دو ناحیه. فرمول آن به صورت دقیق به این شکل است:

$$Dice = (2 * |A \cap B|) / (|A| + |B|)$$

در این فرمول، A نمایانگر ناحیه واقعی یا Ground Truth و B نمایانگر ناحیه پیش‌بینی‌شده توسط مدل است $A \cap B$تعداد پیکسل‌هایی است که هم در ناحیه واقعی و هم در ناحیه پیش‌بینی‌شده وجود دارند (یعنی تعداد پیکسل‌های مشترک)، و Aو Bبه ترتیب تعداد کل پیکسل‌های ناحیه واقعی و ناحیه پیش‌بینی‌شده هستند. مقدار Dice همواره بین صفر و یک قرار می‌گیرد؛ به طوری که اگر مقدار آن برابر با یک باشد، به این معناست که ناحیه پیش‌بینی‌شده و ناحیه واقعی کاملاً با هم تطابق دارند. در مقابل، اگر مقدار Dice برابر صفر باشد، نشان‌دهنده آن است که هیچ همپوشانی بین ناحیه پیش‌بینی‌شده و ناحیه واقعی وجود ندارد. مقادیر بین صفر و یک نیز نشان‌دهنده میزان نسبی همپوشانی بین این دو ناحیه هستند؛ هرچه مقدار Dice به یک نزدیک‌تر باشد، نشان‌دهنده دقت بالاتر مدل است.

برای محاسبه Dice Coefficient در یک تصویر، ابتدا باید دو ماسک در اختیار داشته باشیم؛ ماسک واقعی که ناحیه موردنظر را نشان می‌دهد و ماسک پیش‌بینی‌شده که خروجی مدل است. سپس تعداد پیکسل‌های مشترک بین دو ماسک محاسبه می‌شود. این پیکسل‌ها همان نقاطی هستند که هم توسط مدل به‌درستی شناسایی شده‌اند و هم در واقعیت وجود دارند. پس از آن، مجموع تعداد پیکسل‌های ناحیه واقعی و ناحیه پیش‌بینی‌شده محاسبه می‌شود و در نهایت، Dice برابر است با دو برابر تعداد پیکسل‌های مشترک تقسیم بر مجموع تعداد کل پیکسل‌ها در هر دو ناحیه.

برای مثال، فرض کنید در یک تصویر با اندازه ۱۰ در ۱۰، ناحیه‌ای به‌عنوان Ground Truth داریم که شامل ۲۰ پیکسل است. همچنین مدل ما ناحیه‌ای شامل ۲۵ پیکسل را به‌عنوان پیش‌بینی تشخیص داده است. اگر تعداد پیکسل‌های مشترک بین ناحیه واقعی و ناحیه پیش‌بینی‌شده برابر با ۱۵ باشد، محاسبه Dice به‌صورت زیر خواهد بود:

تعداد پیکسل‌های تقاطع = ۱۵ تعداد کل پیکسل‌های ناحیه واقعی = ۲۰ تعداد کل پیکسل‌های ناحیه پیش‌بینی‌شده = ۲۵

بنابراین:

Dice = (2 * 15) / (20 + 25) = 30 / 45 = 0.666

مقدار Dice برابر با ۰,۶۶۶ است که نشان می‌دهد مدل توانسته ناحیه واقعی را تا حدی با دقت نسبتاً خوبی شناسایی کند.

Dice Coefficient و IoU هر دو معیارهایی برای اندازه‌گیری میزان همپوشانی بین ناحیه پیش‌بینی‌شده و ناحیه واقعی هستند، اما تفاوت‌هایی نیز دارند. ارتباط بین Dice و IoU به صورت ریاضی به این شکل است که:

Dice = (2 * IoU) / (1 + IoU)

IoU = Dice / (2 - Dice)

هر دو معیار اطلاعات مشابهی در مورد میزان همپوشانی ارائه می‌دهند، اما Dice به دلیل فرمولی که دارد، حساسیت بیشتری به تعداد پیکسل‌های مشترک دارد و در بسیاری از موارد که نواحی کوچک هستند یا همپوشانی کمی وجود دارد، Dice می‌تواند نتایج دقیق‌تری ارائه دهد.

یکی از مزایای اصلی Dice Coefficient، حساسیت بالای آن به همپوشانی است. Dice به دلیل استفاده از دو برابر تقاطع، به‌طور مستقیم به تعداد پیکسل‌های مشترک وزن می‌دهد و این موضوع باعث می‌شود که دقت مدل در تشخیص نواحی کوچک و نواحی با تغییرات جزئی بهبود یابد. این ویژگی به‌خصوص در مسائل پزشکی و تصاویر MRI که نواحی مانند تومورها بسیار کوچک و مهم هستند، بسیار مفید است. Dice Coefficient به‌طور گسترده در حوزه پزشکی، به‌ویژه برای ارزیابی دقت مدل‌های سگمنتیشن در شناسایی نواحی بیماری‌زا، مانند تومورهای مغزی، مورد استفاده قرار می‌گیرد.

از دیگر مزایای Dice Coefficient می‌توان به تعادل آن در اندازه‌گیری همپوشانی اشاره کرد. این معیار به‌طور همزمان تعداد پیکسل‌های ناحیه واقعی و ناحیه پیش‌بینی‌شده را در نظر می‌گیرد و این باعث می‌شود که دقت مدل به‌طور تعادل‌یافته‌ای ارزیابی شود. برخلاف معیارهایی که تنها بر یک بخش تأکید می‌کنند، Dice به‌طور کامل به هر دو ناحیه توجه دارد و این امر به‌ویژه در مواردی که توازن بین دو ناحیه اهمیت دارد، مؤثر است.

اما Dice Coefficient معایبی نیز دارد. یکی از معایب این معیار، حساسیت آن به عدم تطابق‌های کوچک است. به این معنا که اگر تعداد پیکسل‌های مشترک کم باشد، مقدار Dice به‌شدت کاهش می‌یابد و این

باعث می‌شود مدل‌هایی که روی نواحی کوچک یا نواحی با تغییرات جزئی عملکرد دارند، نمره پایینی کسب کنند. همچنین، محاسبه Dice در مواردی که اندازه تصویر بزرگ باشد و تعداد زیادی پیکسل برای محاسبه تقاطع و مجموع نیاز باشد، زمان‌بر و پیچیده است.

در یادگیری عمیق، به‌ویژه در مدل‌های شبکه‌های عصبی کانولوشنی که برای مسائل سگمنتیشن تصویر استفاده می‌شوند، Dice Coefficient یکی از معیارهای اصلی برای ارزیابی عملکرد مدل است. این معیار می‌تواند به عنوان تابع هزینه نیز به کار رود. در این حالت، مدل تلاش می‌کند مقدار Dice را به حداکثر برساند تا بتواند همپوشانی بیشتری بین ناحیه واقعی و ناحیه پیش‌بینی‌شده ایجاد کند. استفاده از Dice Coefficient به عنوان تابع هزینه به‌ویژه در مواقعی که کلاس‌های تصویر نابرابر هستند (یعنی تعداد پیکسل‌های یک کلاس بسیار کمتر از دیگری است) مفید است. این مسئله به‌طور خاص در تصاویر پزشکی که تومورها معمولاً ناحیه کوچکی از تصویر را تشکیل می‌دهند، مهم است. در چنین مواردی، استفاده از Dice به عنوان تابع هزینه می‌تواند کمک کند که مدل حساسیت بیشتری نسبت به این نواحی کوچک داشته باشد و بهتر عمل کند.

این دو را پیادهسازی کرده و از آنها به همراه Accuracyبه عنوان معیار حین آموزش شبکه استفاده کنید.

برای پیاده‌سازی معیارهای Dice Coefficient، IoU Score و Accuracy و استفاده از آن‌ها به‌عنوان معیارهای ارزیابی در حین آموزش شبکه، ما مراحل متعددی را طی کردیم تا این معیارها به‌طور مناسب پیاده‌سازی شده و به مدل اعمال شوند. ابتدا تصمیم گرفتیم که این معیارها را به‌عنوان معیارهای کلیدی برای سنجش عملکرد مدل در حین فرایند آموزش به کار بگیریم، زیرا این معیارها قادرند به‌خوبی دقت پیش‌بینی‌های مدل و میزان همپوشانی ناحیه‌های پیش‌بینی‌شده با ناحیه‌های واقعی را نمایش دهند.

در مرحله اول، برای پیاده‌سازی معیار Dice Coefficient، یک تابع به نام dice_coefficient تعریف کردیم. این تابع به‌گونه‌ای طراحی شد که ورودی‌های آن شامل دو پارامتر y_true و y_pred باشد که به ترتیب ماسک واقعی و ماسک پیش‌بینی‌شده توسط مدل هستند. ما ابتدا تصمیم گرفتیم که ورودی‌ها را با استفاده از K.flatten()که از کتابخانه Keras فراخوانی شده، به شکل بردارهای خطی تبدیل کنیم. این کار باعث می‌شد تا محاسبات ریاضی ساده‌تر و سریع‌تر انجام شود. سپس، تعداد پیکسل‌های مشترک بین ماسک واقعی و ماسک پیش‌بینی‌شده را محاسبه کردیم. این تعداد پیکسل‌های مشترک به‌عنوان مقدار intersection در فرمول Dice Coefficient استفاده شد. برای محاسبه مقدار Dice، از فرمول ریاضی آن که بالاتر ذکر شد استفاده کردیم. به‌طور کلی، این معیار به ما این امکان را داد که میزان همپوشانی بین

ناحیه‌های پیش‌بینی‌شده و ناحیه‌های واقعی را بسنجیم. هرچه مقدار Dice بالاتر باشد، نشان‌دهنده این است که مدل توانسته است ناحیه‌های مورد نظر را بهتر شناسایی کند.

سپس، برای پیاده‌سازی معیار IoU Score، تصمیم گرفتیم که تابعی به نام iou_score تعریف کنیم که به مشابه dice_coefficient عمل کند، اما با یک تفاوت عمده در فرمول محاسبه. ما از همان ورودی‌های y_true و y_pred استفاده کردیم و این دو ورودی را به شکل بردارهای خطی تبدیل کردیم. پس از آن، تعداد پیکسل‌های مشترک را به‌عنوان intersection محاسبه کردیم و تعداد کل پیکسل‌های موجود در هر دو ماسک را به‌عنوان union محاسبه کردیم. در نهایت، مقدار IoU را با استفاده از فرمول آن که پیشتر ذکر شد محاسبه کردیم. این معیار نیز به ما کمک می‌کند تا میزان همپوشانی ناحیه‌های پیش‌بینی‌شده و ناحیه‌های واقعی را بسنجیم و به‌طور کلی عملکرد مدل را ارزیابی کنیم. IoU به‌ویژه در مواقعی که نیاز به ارزیابی دقیق‌تر میزان همپوشانی داریم، بسیار مفید است.

ما همچنین معیار Accuracy را به عنوان یکی از معیارهای عمومی ارزیابی مدل در حین آموزش در نظر گرفتیم. Accuracy معیاری است که نشان می‌دهد چند درصد از پیکسل‌ها به‌درستی دسته‌بندی شده‌اند. این معیار به‌ویژه در مواردی که قصد داریم دقت کلی مدل را در نظر بگیریم، بسیار کاربردی است. برای پیاده‌سازی Accuracy، از متریک استاندارد accuracy که در کتابخانه Keras موجود است استفاده کردیم و نیازی به تعریف تابع جداگانه نداشتیم.

هنگامی که این معیارها را پیاده‌سازی کردیم، نوبت به مرحله بعدی یعنی افزودن این معیارها به فرآیند آموزش مدل رسید. برای این کار، از متد model.compile() استفاده کردیم. در این متد، علاوه بر مشخص کردن بهینه‌ساز و تابع هزینه ، معیارهای ارزیابی را نیز به مدل معرفی کردیم. به این ترتیب، معیارهای dice_coefficient و iou_score که خودمان تعریف کرده بودیم و همچنین accuracy را به مدل اضافه کردیم تا در هر epoch از آموزش، مقادیر این معیارها محاسبه و گزارش شوند. این کار باعث شد که در هر مرحله از آموزش، به‌طور همزمان بتوانیم عملکرد مدل را با استفاده از معیارهای مختلف ارزیابی کنیم.

پس از اضافه کردن معیارهای ارزیابی به مدل، از ModelCheckpoint و EarlyStopping به‌عنوان callback استفاده کردیم تا بهینه‌سازی و جلوگیری از overfitting را نیز انجام دهیم. ما معیار val_loss را به عنوان ملاک ذخیره‌سازی بهترین مدل انتخاب کردیم، به این معنی که هر بار که مقدار val_loss بهبود پیدا می‌کرد، مدل ذخیره می‌شد. همچنین، از EarlyStopping برای متوقف کردن آموزش در صورتی که بهبودی در val_loss مشاهده نمی‌شد، استفاده کردیم تا از overfitting جلوگیری کنیم و مدل بهینه‌تری داشته باشیم. در هر epoch از آموزش، مقادیر Dice Coefficient، IoU Score و Accuracy برای داده‌های آموزش و داده‌های اعتبارسنجی محاسبه می‌شدند. این مقادیر به ما امکان می‌داد تا ببینیم

که آیا مدل در حال بهبود است یا خیر و همچنین بفهمیم که مدل چگونه با داده‌های آموزش و داده‌های اعتبارسنجی عمل می‌کند. اگر مشاهده می‌کردیم که مقدار Dice Coefficient یا IoU برای داده‌های اعتبارسنجی نسبت به داده‌های آموزش کاهش یافته است، این به معنی overfitting مدل بود و ما می‌توانستیم اقدامات لازم برای اصلاح مدل را انجام دهیم.

در نهایت، هدف ما از استفاده از این معیارها این بود که بتوانیم به‌طور جامع و دقیق عملکرد مدل را ارزیابی کنیم. معیار Dice Coefficient به ما کمک می‌کرد تا میزان همپوشانی بین ناحیه‌های واقعی و ناحیه‌های پیش‌بینی‌شده را بسنجیم و بفهمیم که مدل تا چه اندازه به‌درستی توانسته است ناحیه‌های مورد نظر را شناسایی کند. معیار IoU نیز به‌طور مشابه به ما کمک می‌کرد تا میزان همپوشانی را ارزیابی کنیم، اما از نظر فرمول محاسباتی تفاوت‌هایی با Dice داشت که باعث می‌شد حساسیت بیشتری نسبت به اندازه ناحیه‌ها داشته باشد. Accuracy نیز به عنوان یک معیار کلی، به ما نشان می‌داد که مدل تا چه اندازه توانسته است پیکسل‌ها را به‌درستی دسته‌بندی کند.

```python
def dice_coefficient(y_true, y_pred, smooth=1):
    y_true_f = K.flatten(y_true)
    y_pred_f = K.flatten(y_pred)
    intersection = K.sum(y_true_f * y_pred_f)
    return (2. * intersection + smooth) / (K.sum(y_true_f) + K.sum(y_pred_f) + smooth)

def iou_score(y_true, y_pred, smooth=1):
    y_true_f = K.flatten(y_true)
    y_pred_f = K.flatten(y_pred)
    intersection = K.sum(y_true_f * y_pred_f)
    union = K.sum(y_true_f) + K.sum(y_pred_f) - intersection
    return (intersection + smooth) / (union + smooth)

def combined_dice_bce_loss(y_true, y_pred):
    bce = tf.keras.losses.BinaryCrossentropy()(y_true, y_pred)
    dice = 1 - dice_coefficient(y_true, y_pred)
    return bce + dice

def dice_loss(y_true, y_pred):
    return 1 - dice_coefficient(y_true, y_pred)
```

شکل ۶پیاده سازی های مربوط به متریک های ارزیابی سگمنتیشن

بهینه ساز و تابع هزینه را مطابق با مقاله یا با انتخاب خودتان تنظیم کرده و آنها را گزارش کنید.

پیشنهاد میشود برای تابع هزینه از Dice lossاستفاده کنید

برای تنظیم بهینه‌ساز و تابع هزینه در این پروژه، تصمیم گرفتیم که از بهینه‌ساز Adam و ترکیبی از Dice Loss و Binary Cross-Entropy (BCE) به عنوان تابع هزینه استفاده کنیم. این تصمیم بر اساس بررسی دقیق نیازهای مسئله و ویژگی‌های هر یک از این توابع و همچنین پیشنهاد مقاله‌ای که برای این پروژه مرجع قرار داده شده بود، اتخاذ شد. در این بخش، جزئیات تصمیم‌گیری‌ها و نحوه پیاده‌سازی را با دقت توضیح می‌دهیم. ابتدا برای انتخاب بهینه‌ساز، بهینه‌ساز Adam را انتخاب کردیم. دلیل این انتخاب، سرعت بالای بهینه‌ساز Adam در همگرایی و همچنین توانایی آن در مدیریت گرادیان‌های بزرگ و به‌روزرسانی پارامترها به شکل تطبیقی است. بهینه‌ساز Adam با ترکیب مزایای روش‌های RMSProp و Momentum، سرعت همگرایی را افزایش داده و درعین‌حال پایدار بودن آموزش مدل را تضمین می‌کند. این بهینه‌ساز به‌ویژه در مسائل پیچیده مانند شبکه‌های عصبی کانولوشنی که تعداد پارامترهای زیادی دارند و حجم داده‌های ورودی بالا است، بسیار کارآمد است. در متد ()model.compile، بهینه‌ساز Adam با نرخ یادگیری learning_rate=1e-4 تنظیم شد. انتخاب نرخ یادگیری مناسب از اهمیت بالایی برخوردار است، زیرا نرخ یادگیری بالا ممکن است باعث نوسان‌های زیاد در فرآیند آموزش و نرسیدن به بهترین نقطه بهینه شود، در حالی که نرخ یادگیری خیلی کوچک می‌تواند فرآیند همگرایی را بسیار کند کند. با توجه به نیاز پروژه و تجربیات گذشته، نرخ یادگیری ۱e-4 برای این کار مناسب تشخیص داده شد.

در بخش بعدی، تصمیم گرفتیم که تابع هزینه را به صورت ترکیبی از Dice Loss و Binary Cross-Entropy انتخاب کنیم. پیشنهاد شده بود که از Dice Loss برای تابع هزینه استفاده شود، زیرا Dice Loss به‌طور مستقیم به هدف مسئله سگمنتیشن مرتبط است و تلاش می‌کند تا میزان همپوشانی بین ناحیه‌های پیش‌بینی‌شده و ناحیه‌های واقعی را به حداکثر برساند. Dice Loss به‌ویژه در مسائل سگمنتیشن با چالش‌های عدم تعادل بین پیکسل‌های ناحیه هدف و پیکسل‌های پس‌زمینه بسیار مفید است، چرا که این عدم تعادل باعث می‌شود مدل تمایل داشته باشد تا بیشتر پیکسل‌ها را به پس‌زمینه تخصیص دهد و ناحیه‌های کوچک‌تر را نادیده بگیرد. Dice Loss با تمرکز بر میزان همپوشانی نواحی، به مدل کمک می‌کند تا این مشکل را کاهش دهد و به درستی ناحیه‌های کوچک‌تر را نیز شناسایی کند.

برای پیاده‌سازی Dice Loss، تابع dice_coefficient که پیش‌تر توضیح داده شد، مورد استفاده قرار گرفت. سپس با استفاده از این تابع، Dice Loss را به شکل ۱ - dice_coefficient(y_true, y_pred) محاسبه کردیم. این تغییر باعث شد که تابع Dice به جای اینکه حداکثر شود، به عنوان تابع هزینه‌ای که باید به حداقل برسد، عمل کند. هدف در اینجا این است که مدل بتواند همپوشانی بین ماسک پیش‌بینی‌شده و ماسک واقعی را به بیشترین حد ممکن برساند که این امر با حداقل کردن Dice Loss حاصل می‌شود.

در کنار Dice Loss، از Binary Cross-Entropy (BCE) نیز به عنوان بخشی از تابع هزینه استفاده کردیم. استفاده از ترکیب BCE و Dice Loss به این دلیل انجام شد که BCE به‌خوبی به مدل کمک می‌کند تا تفاوت‌های جزئی بین ماسک واقعی و ماسک پیش‌بینی‌شده را شناسایی کند و همچنین در بهبود نرخ همگرایی نیز مؤثر است. BCE به عنوان یک معیار رایج در مسائل باینری (دودویی) کمک می‌کند که مدل به‌خوبی تفاوت بین پیکسل‌هایی که باید به عنوان ناحیه تومور یا ناحیه پس‌زمینه شناخته شوند، را شناسایی کند.

در نهایت، تابع هزینه‌ای که برای آموزش مدل انتخاب شد، ترکیبی از Dice Loss و BCE بود که در کد به صورت combined_dice_bce_loss پیاده‌سازی شد. این تابع هزینه به این شکل تعریف شد که ابتدا مقدار BCE با استفاده از تابع tf.keras.losses.BinaryCrossentropy()محاسبه می‌شود و سپس Dice Loss با استفاده از ۱ - dice_coefficient(y_true, y_pred) به دست می‌آید. در نهایت، مجموع این دو به عنوان تابع هزینه استفاده شد. این ترکیب به این دلیل انتخاب شد که بتواند به‌طور همزمان هم میزان همپوشانی را افزایش دهد و هم تفاوت‌های جزئی بین ماسک‌ها را به‌درستی تشخیص دهد.

با ترکیب بهینه‌ساز Adam و این تابع هزینه ترکیبی، مدل ما به گونه‌ای آموزش دید که بتواند به‌طور بهینه‌تری ناحیه‌های تومور را در تصاویر MRI شناسایی کند. بهینه‌ساز Adam به ما کمک کرد تا فرآیند آموزش با سرعت بیشتری همگرا شود و مشکلات مربوط به گرادیان‌های بزرگ و ناپایدار را مدیریت کند. از سوی دیگر، استفاده از ترکیب Dice Loss و BCE به مدل کمک کرد که به‌خوبی با مشکلات عدم تعادل داده‌ها مقابله کرده و ناحیه‌های مورد نظر را با دقت بیشتری شناسایی کند. این ترکیب بهینه از بهینه‌ساز و تابع هزینه، ما را قادر ساخت تا مدلی بسازیم که عملکرد بالایی در تشخیص ناحیه‌های تومور داشته باشد و نتایج حاصل از آموزش نشان دادند که این انتخاب‌ها به بهبود دقت و همپوشانی مدل کمک شایانی کرده‌اند.

۱,۵-پیاده سازی مدل

برای پیاده‌سازی مدل UNet-VGG16 جهت حل مسئله سگمنتیشن تصاویر، ابتدا نیاز به استفاده از شبکه VGG16 به‌عنوان بخش Encoder مدل داشتیم، سپس لایه‌های مخصوص Decoder به مدل اضافه شدند تا ساختار کامل UNet به دست آید. در این گزارش، گام‌های پیاده‌سازی و انتخاب‌های ما در هر بخش توضیح داده می‌شود.

ابتدا تصمیم گرفتیم که از VGG16 به‌عنوان یک Encoder در مدل UNet استفاده کنیم، زیرا VGG16 به‌عنوان یک مدل از پیش آموزش‌دیده روی مجموعه داده ImageNet، قابلیت استخراج ویژگی‌های قوی را دارد. این موضوع به ما کمک کرد که بتوانیم از ویژگی‌های یادگرفته‌شده این مدل استفاده کنیم و فرایند یادگیری را تسریع کنیم. ما از وزن‌های از پیش آموزش‌دیده شده imagenet در VGG16 بهره بردیم تا بتوانیم از اطلاعات به‌دست آمده از آن استفاده کنیم و قابلیت‌های آن را در بخش اولیه مدل خود به‌کار ببریم. استفاده از VGG16 به‌عنوان بخش Encoder باعث شد که مدل توانایی بیشتری در استخراج ویژگی‌های پیچیده از تصاویر ورودی داشته باشد.

برای پیاده‌سازی این مرحله، ما از VGG16 موجود در کتابخانه tensorflow.keras.applications استفاده کردیم و آن را به صورت include_top=False بارگذاری کردیم تا لایه‌های Fully Connected نهایی حذف شوند، زیرا این لایه‌ها برای کار ما لازم نبودند و هدف ما استخراج ویژگی‌های سطح پایین و میانی از تصاویر بود. همچنین ورودی مدل VGG16 را با اندازه‌ی (۲۵۶, ۲۵۶, ۳) تنظیم کردیم تا با اندازه تصاویر ورودی سازگار باشد.

بعد از بارگذاری VGG16، لایه‌های بخش Encoder را برای پیاده‌سازی مدل UNet به صورت دقیق استخراج کردیم. لایه‌های مشخصی از مدل VGG16، نظیر block1_conv2، block2_conv2، block3_conv3، و block4_conv3 به‌عنوان نقاط اتصال استفاده شدند که در فرایند Decoder برای انجام عملیات Skip Connection از آن‌ها بهره گرفتیم. این لایه‌ها به ما کمک می‌کنند که اطلاعات سطح پایین و جزئیات تصاویر را از بخش Encoder به بخش Decoder منتقل کنیم، که این خود باعث می‌شود کیفیت خروجی بهبود یابد.

برای بخش Decoder مدل، عملیات UpSampling را با استفاده از لایه‌های UpSampling2D انجام دادیم. در هر مرحله از Decoder، ابتدا ویژگی‌های به‌دست آمده از لایه‌های Encoder را با لایه‌های UpSampling2D ترکیب کردیم. این ترکیب با استفاده از عملیات concatenate انجام شد که به Skip Connections معروف است. در واقع، هدف از این کار، بازگرداندن اطلاعات از دست‌رفته در طول مراحل DownSampling و حفظ جزئیات و دقت در هنگام بازسازی تصویر بود.

هر یک از لایه‌های UpSampling، پس از ترکیب ویژگی‌ها با لایه‌های Skip Connection، به چندین لایه کانولوشن متصل شدند. به این ترتیب، ما از لایه‌های Conv2D برای استخراج ویژگی‌های جدید از ترکیب اطلاعات و لایه‌های BatchNormalization برای نرمال‌سازی و کاهش پراکندگی استفاده کردیم. به علاوه، در تمامی مراحل از لایه‌های فعال‌سازی LeakyReLU استفاده شد تا عملیات غیرخطی‌سازی به‌خوبی

اعمال شود و شبکه توانایی بیشتری در مدل‌سازی روابط غیرخطی داشته باشد. در هر مرحله از بخش Decoder، برای جلوگیری از overfitting، از لایه‌های Dropout نیز استفاده کردیم.

در نهایت، لایه آخر Decoder شامل یک لایه کانولوشن Conv2D با فیلتر خروجی به تعداد ۱ بود که با فعال‌سازی sigmoid تنظیم شد. این لایه خروجی، نقشه احتمالاتی برای سگمنتیشن تصویر تولید می‌کند که هر پیکسل دارای مقداری بین ۰ و ۱ است، و نشان می‌دهد که احتمال تعلق آن پیکسل به ناحیه تومور چقدر است. هدف ما این بود که با استفاده از این ساختار، مدلی داشته باشیم که بتواند ناحیه‌های تومور را به‌طور دقیق شناسایی کرده و خروجی ماسک مورد نظر را تولید کند.

برای آموزش مدل UNet-VGG16، از بهینه‌ساز Adam با نرخ یادگیری ۰,۰۰۱ استفاده کردیم که به‌طور قبلی توضیح دادیم که این انتخاب به دلیل توانایی Adam در به‌روزرسانی‌های سریع و مدیریت گرادیان‌های بزرگ بود. همچنین، برای ارزیابی عملکرد مدل و بهینه‌سازی، از ترکیبی از Dice Loss و Binary Cross-Entropy استفاده شد تا هم بتوانیم نواحی کوچک‌تر را بهتر شناسایی کنیم و هم دقت پیش‌بینی‌ها را بهبود ببخشیم. این ترکیب باعث شد که مدل توانایی بهتری در شناسایی و تفکیک ناحیه‌های تومور داشته باشد.

در طول آموزش مدل، از دو callback به نام‌های ModelCheckpoint و EarlyStopping استفاده کردیم. ModelCheckpoint برای ذخیره بهترین مدل در طول آموزش و EarlyStopping برای جلوگیری از overfitting و توقف آموزش در صورتی که بهبودی در عملکرد مدل مشاهده نشد، به‌کار رفتند. این اقدامات باعث شد که مدل بهینه‌تری در پایان فرایند آموزش داشته باشیم و از آموزش اضافی که ممکن بود باعث overfitting شود، جلوگیری کنیم.

۱,۶- آموزش مدل

برای آموزش مدل UNet-VGG16، تصمیم گرفتیم که تعداد epochها و batch size را با دقت انتخاب کنیم تا فرایند آموزش مدل بهینه و در عین حال کارآمد باشد. در این بخش، به‌طور دقیق توضیح می‌دهیم که چگونه این پارامترها را انتخاب کردیم و چرا این انتخاب‌ها به بهبود آموزش مدل کمک کرده‌اند.

ابتدا تعداد epochها را تعیین کردیم. هدف از انتخاب تعداد مناسب epoch این بود که مدل به‌خوبی یاد بگیرد و از سوی دیگر از مشکلاتی مانند overfitting جلوگیری شود. به این منظور، تصمیم گرفتیم که تعداد epochها را به‌صورت ۳۰ انتخاب کنیم. دلیل انتخاب این تعداد این بود که بر اساس تجربه‌های قبلی و انجام چند آزمایش اولیه، متوجه شدیم که در این تعداد epoch، مدل می‌تواند به‌خوبی ویژگی‌های مورد نظر را از تصاویر استخراج کند و به همگرایی مطلوبی برسد. اگر تعداد epochها خیلی کم باشد، ممکن

است مدل هنوز به‌خوبی یاد نگرفته باشد و ویژگی‌های پیچیده تصاویر را شناسایی نکند، در نتیجه با مشکل underfitting مواجه می‌شویم. از طرفی، تعداد زیاد epochها می‌تواند منجر به overfitting شود، که در این حالت مدل به‌شدت روی داده‌های آموزشی منطبق می‌شود و عملکرد آن روی داده‌های جدید کاهش می‌یابد.

برای جلوگیری از overfitting و بهبود کارایی، از callback به نام EarlyStopping نیز استفاده کردیم. این callback با تنظیم پارامتر patience به ۱۰، به مدل اجازه داد که اگر بهبودی در مقدار val_loss دیده نشد، آموزش را متوقف کند. این کار به ما کمک کرد که بتوانیم تعداد epochها را به‌صورت دینامیک تنظیم کنیم و در صورتی که مدل دیگر بهبود نیافت، از آموزش بیشتر جلوگیری شود. این امر به‌خصوص در صرفه‌جویی در منابع محاسباتی و جلوگیری از زمان اضافی آموزش مؤثر بود.

سپس به انتخاب batch size پرداختیم. batch size، تعداد نمونه‌هایی است که در هر مرحله از به‌روزرسانی پارامترها در طول آموزش مدل استفاده می‌شود. در این پروژه، تصمیم گرفتیم که مقدار batch size را برابر با ۱۶ انتخاب کنیم. این مقدار به دلیل متوازن بودن بین دو جنبه انتخاب شد: اول اینکه batch size کوچک‌تر می‌تواند به مدل کمک کند تا گرادیان‌های دقیق‌تری محاسبه کند و به سرعت همگرایی بالاتری برسد، اما از طرفی منابع محاسباتی بیشتری مصرف می‌کند و زمان آموزش ممکن است طولانی‌تر شود. batch size بزرگ‌تر می‌تواند زمان آموزش را کاهش دهد اما ممکن است گرادیان‌ها به‌طور دقیق به‌روزرسانی نشوند و مشکلات نوسان در فرآیند آموزش به‌وجود آید. با انتخاب batch size برابر ۱۶، توانستیم تعادلی بین دقت آموزش و استفاده بهینه از منابع محاسباتی ایجاد کنیم.

فرایند آموزش مدل با تعداد epoch برابر با ۳۰ و batch size برابر ۱۶ انجام شد. مدل با استفاده از داده‌های آموزشی و اعتبارسنجی آموزش دید و عملکرد آن در هر epoch مورد ارزیابی قرار گرفت. برای هر epoch، مقادیر مربوط به loss و متریک‌های dice_coefficient و iou_score برای داده‌های آموزشی و اعتبارسنجی ثبت شد. هدف ما این بود که با بررسی این مقادیر، بهبود عملکرد مدل را در طول آموزش پیگیری کنیم.

رسم نمودار و تحلیل آن

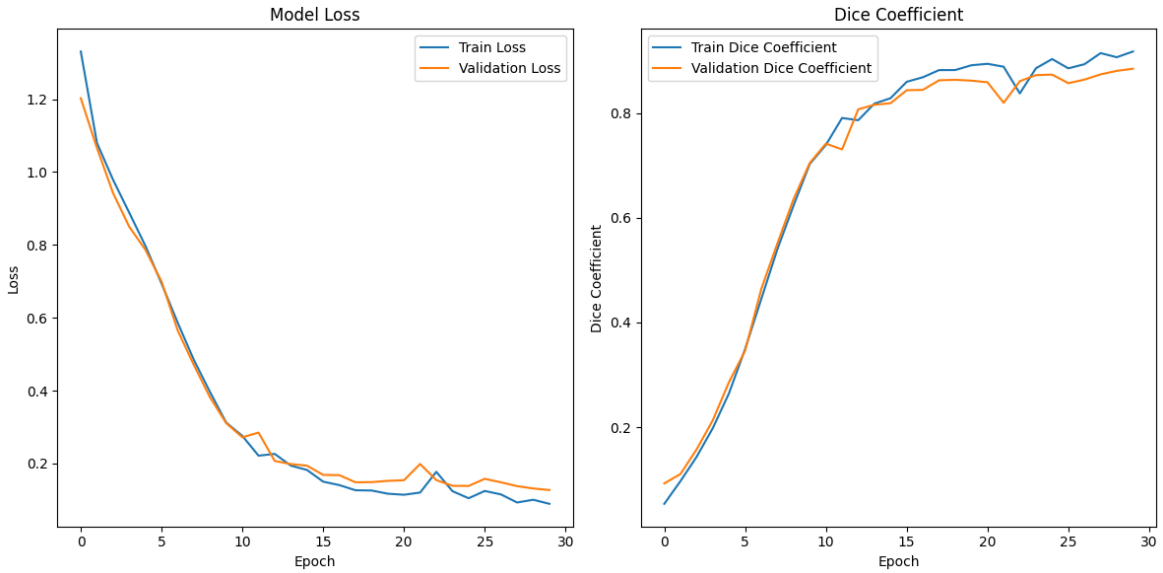
پس از آموزش مدل، تصمیم گرفتیم که برای ارزیابی عملکرد آن و مشاهده روند تغییرات در طول epochها، نمودارهای مربوط به متریک‌های مختلف و تابع هزینه را روی داده‌های آموزش و ارزیابی رسم کنیم. این نمودارها به ما کمک کردند تا بتوانیم به‌صورت بصری روند بهبود مدل را مشاهده کنیم و مشخص کنیم که مدل چگونه در هر مرحله از آموزش و روی داده‌های اعتبارسنجی عمل کرده است. در این گزارش، جزئیات گام‌های انجام‌شده برای رسم نمودارها و تحلیل آن‌ها ارائه می‌شود. ابتدا تصمیم گرفتیم که مقدار تابع هزینه

را برای داده‌های آموزشی و ارزیابی در طول epochها رسم کنیم. این کار به این دلیل انجام شد که بتوانیم به‌صورت بصری مشاهده کنیم که آیا مدل به درستی به حداقل مقدار تابع هزینه همگرا شده است یا خیر. برای این منظور، از تاریخچه آموزش که در متغیر history ذخیره شده بود استفاده کردیم. history.history['loss'] مقادیر مربوط به تابع هزینه برای داده‌های آموزشی و history.history['val_loss'] مقادیر مربوط به تابع هزینه برای داده‌های اعتبارسنجی را در هر epoch در اختیار ما قرار می‌داد.

ما نمودار تغییرات loss را برای داده‌های آموزشی و ارزیابی در یک زیرنمودار رسم کردیم. این نمودار به ما نشان داد که با گذشت زمان، مقدار تابع هزینه روی داده‌های آموزشی به‌صورت یکنواخت کاهش یافته است، که این نشانه خوبی از یادگیری مدل است. از سوی دیگر، مقدار تابع هزینه روی داده‌های اعتبارسنجی نیز کاهش یافته و به یک مقدار ثابت همگرا شده است. این موضوع نشان می‌دهد که مدل به‌خوبی یاد گرفته است و همچنین به overfitting دچار نشده است، زیرا تفاوت بین loss آموزش و اعتبارسنجی زیاد نبوده و هر دو به یک مقدار مشابه همگرا شده‌اند. در صورتی که نمودار loss اعتبارسنجی به‌جای کاهش، افزایش می‌یافت یا دچار نوسانات زیاد می‌شد، این موضوع نشان‌دهنده overfitting بود که خوشبختانه چنین مشکلی در نتایج ما مشاهده نشد.

سپس تصمیم گرفتیم که متریک‌های Dice Coefficient را که برای ارزیابی دقت مدل استفاده شده بودند، نیز رسم کنیم. این متریک‌ها برای ما اهمیت ویژه‌ای داشتند، چرا که مستقیماً به میزان دقت و کیفیت سگمنتیشن مدل اشاره دارند. Dice Coefficient میزان همپوشانی بین ناحیه‌های پیش‌بینی‌شده و ناحیه‌های واقعی را اندازه‌گیری می‌کند، و IoU Score یا Intersection over Union نیز معیار دیگری برای ارزیابی همپوشانی بین ناحیه‌های پیش‌بینی‌شده و واقعی است. این دو متریک نشان می‌دهند که مدل تا چه حد در شناسایی ناحیه‌های تومور موفق بوده است.

برای رسم این متریک‌ها، ابتدا مقادیر Dice Coefficient را برای داده‌های آموزشی و اعتبارسنجی از تاریخچه آموزش استخراج کردیم. history.history['dice_coefficient'] نشان‌دهنده مقادیر Dice Coefficient برای داده‌های آموزشی و history.history['val_dice_coefficient'] نشان‌دهنده این متریک برای داده‌های اعتبارسنجی بود. نمودار این مقادیر نشان داد که Dice Coefficient در طول epochها به‌صورت یکنواخت افزایش یافته و به مقدار بالایی همگرا شده است، که این موضوع نشان‌دهنده افزایش دقت مدل در شناسایی ناحیه‌های تومور است. همچنین مشاهده کردیم که مقادیر Dice Coefficient برای داده‌های آموزشی و اعتبارسنجی به‌صورت مشابهی افزایش یافته‌اند، که نشان‌دهنده این است که مدل به‌خوبی یاد گرفته است و تعمیم‌پذیری مناسبی دارد.



شکل ۷نمودار تغییرات خط و متریک ها در طول ایپاک ها

ما این نمودارها را رسم کردیم تا عملکرد مدل در طول فرآیند آموزش را به‌دقت تحلیل کنیم. نمودار اول تغییرات تابع هزینه را برای داده‌های آموزشی و اعتبارسنجی نشان می‌دهد. در ابتدای آموزش، مقدار loss برای هر دو مجموعه داده بسیار بالا بود که نشان‌دهنده ناتوانی مدل در تشخیص ناحیه‌های تومور در شروع یادگیری است. با افزایش تعداد epochها، loss به‌صورت پیوسته کاهش یافت و به تدریج به یک مقدار پایین و ثابت رسید. این کاهش همزمان در loss برای داده‌های آموزشی و اعتبارسنجی نشان‌دهنده یادگیری خوب و پیوسته مدل است. نکته‌ای که مشاهده کردیم این بود که مقدار loss برای داده‌های اعتبارسنجی تقریباً به‌طور همگام با داده‌های آموزشی کاهش می‌یافت، که نشان‌دهنده نبود مشکل overfitting است و مدل توانایی تعمیم‌پذیری خوبی داشت.

نمودار دوم مربوط به تغییرات Dice Coefficient است که یکی از معیارهای اصلی برای سنجش دقت مدل در مسائل سگمنتیشن است. این معیار میزان همپوشانی بین ناحیه‌های پیش‌بینی شده توسط مدل و ناحیه‌های واقعی را اندازه‌گیری می‌کند. در ابتدا مقدار Dice Coefficient بسیار پایین بود، که نشان می‌داد مدل در شروع یادگیری توانایی کافی برای تشخیص نواحی تومور را ندارد. اما با افزایش تعداد epochها، مقدار Dice Coefficient به‌تدریج افزایش یافت و به مقادیر بالایی نزدیک به ۰,۹ رسید. این روند صعودی در هر دو مجموعه داده آموزشی و اعتبارسنجی نشان داد که مدل توانسته است به‌صورت کلی بهبود قابل توجهی در عملکرد خود ایجاد کند و توانایی تعمیم‌دهی مناسبی به داده‌های جدید پیدا کرده است.

یکی از نکات قابل توجه در این نمودارها، رابطه میان تغییرات loss و Dice Coefficient بود. با کاهش مقدار loss، مقدار Dice Coefficient به‌طور همزمان افزایش یافت، که این نشان‌دهنده بهبود عملکرد مدل در کاهش خطا و افزایش دقت در شناسایی نواحی تومور بود. این تطابق نشان داد که بهینه‌سازی مدل به‌درستی انجام شده و هرچه loss کاهش می‌یابد، دقت مدل نیز بهبود پیدا می‌کند.

در ابتدای آموزش، loss به شدت کاهش یافته و Dice Coefficient به سرعت افزایش یافت. این نشان‌دهنده آن بود که مدل در مراحل اولیه به سرعت ویژگی‌های مهم را از داده‌ها یاد می‌گیرد و پیشرفت قابل توجهی در عملکرد خود دارد. در epochهای میانی، کاهش loss و افزایش Dice Coefficient یکنواخت‌تر و با سرعت کمتری انجام شد که نشان‌دهنده نزدیک شدن مدل به حالت بهینه بود. در نهایت، در epochهای انتهایی هر دو نمودار به مقادیر ثابتی رسیدند که نشان‌دهنده رسیدن مدل به یک تعادل میان دقت و خطا بود.

به‌طور کلی، این نمودارها به ما اطمینان دادند که مدل به‌صورت موفقیت‌آمیزی آموزش دیده است. مقدار loss کاهش یافت و به حداقل رسید، در حالی که Dice Coefficient به مقدار بالایی نزدیک به ۹.۰ رسید که نشان‌دهنده دقت بالای مدل در شناسایی نواحی تومور است. رفتار مشابه متریک‌ها روی داده‌های آموزشی و اعتبارسنجی نشان داد که مدل دچار overfitting نشده و توانایی تعمیم‌دهی خوبی دارد. این نتایج به ما اطمینان داد که مدل نهایی نه تنها در داده‌های آموزشی، بلکه روی داده‌های جدید نیز می‌تواند نواحی تومور را با دقت بالا شناسایی کند.

در نهایت نتایج زیر از آموزش مدل بدست آمد:

```
accuracy: 0.9986 - dice_coefficient: 0.9144 - iou_score: 0.8449 - loss: 0.0922
- val_accuracy: 0.9980 - val_dice_coefficient: 0.8845 - val_iou_score: 0.7980
- val_loss: 0.1273
```

ما نتایج مدل را پس از پایان آموزش مورد تحلیل قرار دادیم و به نتایج زیر رسیدیم که نشان‌دهنده عملکرد مدل در تشخیص نواحی تومور مغزی است. ابتدا باید به معیار accuracy بپردازیم که برای داده‌های آموزشی و اعتبارسنجی به ترتیب برابر با ۰,۹۹۸۶ و ۰,۹۹۸۰ است. این مقدار بالا نشان می‌دهد که مدل توانسته است تقریباً تمام پیکسل‌ها را به‌درستی تشخیص دهد. با این حال، در مسائل سگمنتیشن تنها accuracy به تنهایی معیار خوبی نیست، زیرا می‌تواند تحت تأثیر تعداد زیاد پیکسل‌های پس‌زمینه قرار گیرد.

معیار مهم‌تر برای ما در اینجا dice_coefficient است که به دقت نشان می‌دهد چقدر نواحی پیش‌بینی شده با نواحی واقعی تطابق دارند. مقدار dice_coefficient برای داده‌های آموزشی برابر با ۰,۹۱۴۴ و برای داده‌های اعتبارسنجی برابر با ۰,۸۸۴۵ به‌دست آمد. این مقدار نشان‌دهنده یک همپوشانی خوب بین نواحی پیش‌بینی شده و نواحی واقعی است و تأیید می‌کند که مدل توانسته است تومور را با دقت خوبی شناسایی کند. کاهش جزئی در مقدار Dice Coefficient برای داده‌های اعتبارسنجی نسبت به داده‌های آموزشی، قابل انتظار است و به‌طور کلی نشان‌دهنده این است که مدل به‌طور مناسبی در حال تعمیم‌دهی به داده‌های جدید است، بدون آنکه دچار overfitting شده باشد.

معیار iou_score نیز برای ارزیابی میزان همپوشانی نواحی پیش‌بینی شده با نواحی واقعی استفاده می‌شود. مقدار iou_score برای داده‌های آموزشی برابر با ۰,۸۴۴۹ و برای داده‌های اعتبارسنجی برابر با ۰,۷۹۸۰ بود. این نتایج نشان می‌دهند که مدل ما به‌خوبی توانسته است نواحی تومور را شناسایی کند و درصد بزرگی از پیکسل‌های ناحیه واقعی با پیش‌بینی‌های مدل همپوشانی دارند. کاهش مختصر در مقدار iou_score برای داده‌های اعتبارسنجی نسبت به داده‌های آموزشی نیز قابل انتظار است، زیرا مدل همیشه در داده‌های آموزشی بهتر عمل می‌کند. اما همچنان این مقدار بالا نشان‌دهنده عملکرد مناسب مدل در داده‌های جدید است.

همچنین به معیار loss نیز دقت کردیم که برای داده‌های آموزشی برابر با ۰,۰۹۲۲ و برای داده‌های اعتبارسنجی برابر با ۰,۱۲۷۳ بود. مقدار loss کمتر نشان‌دهنده کاهش خطا در پیش‌بینی مدل است. اختلاف اندکی که میان loss داده‌های آموزشی و اعتبارسنجی مشاهده شد، نشان‌دهنده این است که مدل به‌طور مناسبی در حال یادگیری ویژگی‌های مهم داده‌هاست و دچار overfitting نشده است. اگرچه مقدار loss برای داده‌های اعتبارسنجی کمی بیشتر از داده‌های آموزشی است، اما این مقدار همچنان به‌اندازه‌ای پایین است که نشان می‌دهد مدل عملکرد مناسبی در تعمیم‌دهی به داده‌های جدید دارد.

۱,۷- ارزیابی مدل

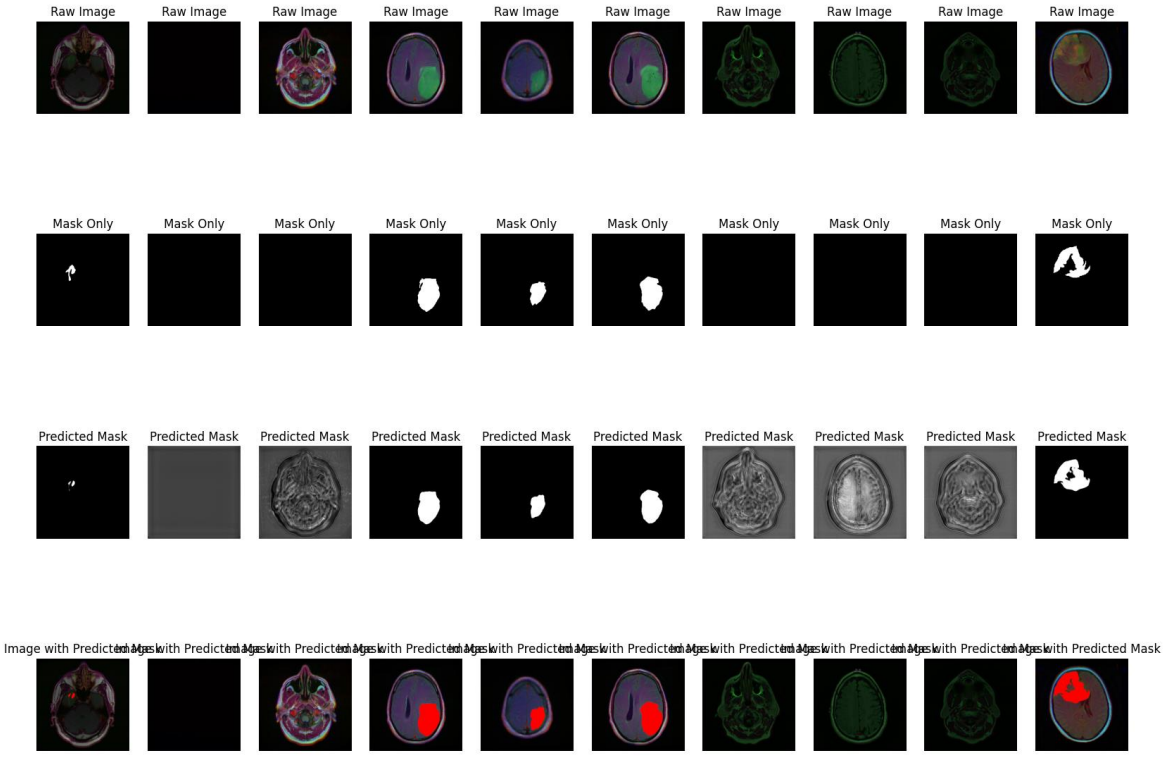
در این مرحله، برای هر تصویر، سه نسخه مختلف از آن را به نمایش گذاشتیم. اولین نسخه، تصویر خام MRI بود که در واقع همان داده اولیه‌ای است که مدل باید آن را تحلیل کند. نسخه دوم، ماسک واقعی (ground truth) تومور بود که به‌صورت دستی توسط متخصصین ایجاد شده و نواحی تومور در آن مشخص شده‌اند. این ماسک به ما کمک می‌کند که بفهمیم نواحی تومور در تصویر اصلی کجا قرار دارند و معیاری

برای ارزیابی پیش‌بینی‌های مدل باشد. نسخه سوم، ماسک پیش‌بینی شده توسط مدل ما بود که خروجی مدل را نشان می‌دهد و به ما می‌گوید که مدل تشخیص داده است کدام نواحی شامل تومور می‌باشند.

در مقایسه بصری این سه نسخه، مشاهده کردیم که مدل ما در بیشتر موارد توانسته است نواحی تومور را به‌درستی شناسایی کند. ماسک‌های پیش‌بینی شده بسیار مشابه ماسک‌های واقعی بودند و در بیشتر موارد تطابق خوبی بین آن‌ها وجود داشت. مدل توانسته بود به‌طور دقیق نواحی تومور را تشخیص داده و این نواحی را با رنگ قرمز بر روی تصویر مشخص کند. همچنین، در مواردی که تومور در ابعاد کوچکتری بود یا به‌صورت غیر منظم در تصویر پخش شده بود، مدل به‌خوبی عمل کرد و به طور کلی توانست الگوهای پیچیده‌تر تومور را نیز شناسایی کند.

با این حال، در برخی نمونه‌ها مشاهده کردیم که ماسک‌های پیش‌بینی شده کاملاً مشابه ماسک‌های واقعی نبودند. در برخی موارد، مدل توانسته بود تنها بخشی از ناحیه تومور را شناسایی کند، یا به‌عبارتی قسمت‌هایی از تومور را نادیده گرفته بود. همچنین، در مواردی دیگر مدل نواحی کوچکی را به اشتباه به‌عنوان تومور مشخص کرده بود که در ماسک واقعی وجود نداشت. این مشکلات عمدتاً به دلیل شباهت بافت‌های نرمال مغز به بافت‌های تومور و پیچیدگی‌های ذاتی تصاویر MRI ایجاد شده‌اند.

با توجه به نتایج به‌دست‌آمده، مشخص شد که مدل ما قادر است نواحی تومور را با دقت بالایی شناسایی کند و در بسیاری از موارد توانسته است که به‌درستی مناطق موردنظر را پیش‌بینی کند. همچنین، تطابق خوب ماسک‌های پیش‌بینی شده با ماسک‌های واقعی نشان داد که مدل به‌خوبی یاد گرفته است تا ویژگی‌های مهم و متمایز تومور را در تصاویر تشخیص دهد. نتایج نشان داد که مدل در تشخیص تومورهای بزرگتر و مشخص عملکرد بهتری دارد، اما در تشخیص تومورهای کوچک‌تر یا پیچیده‌تر ممکن است دچار اشتباه شود.



شکل ۸ نمونه هایی از پیش بینی های انجام شده روی دادگان تست

در اولین ردیف، تصاویر خام MRI نشان داده شده‌اند. این تصاویر همان داده‌های اصلی هستند که مدل بر روی آن‌ها آموزش دیده و حالا باید بر اساس آن‌ها نواحی تومور را پیش‌بینی کند. در این تصاویر می‌توان مشاهده کرد که داده‌های MRI مغز شامل پیچیدگی‌های زیادی هستند و نواحی مختلفی از مغز به‌خوبی قابل تشخیص هستند.

در ردیف دوم، ماسک‌های واقعی نمایش داده شده‌اند که توسط متخصصین برای نشانه‌گذاری نواحی تومور تهیه شده‌اند. این ماسک‌ها به‌صورت باینری (سیاه و سفید) بوده و نشان می‌دهند که کدام نواحی از تصویر مغزی شامل تومور هستند. مشاهده می‌شود که در برخی از تصاویر، تومورها بزرگتر بوده و به‌صورت واضح قابل تشخیص هستند، در حالی که در برخی دیگر، تومورهای کوچکتر و کمتر مشخص دیده می‌شوند.

در ردیف سوم، ماسک‌های پیش‌بینی شده توسط مدل نمایش داده شده‌اند. این پیش‌بینی‌ها نشان می‌دهند که مدل ما کدام نواحی را به‌عنوان تومور شناسایی کرده است. در بررسی این ماسک‌ها مشاهده می‌شود که مدل در بیشتر موارد توانسته نواحی تومور را به‌درستی شناسایی کند و پیش‌بینی‌های آن با ماسک‌های واقعی تطابق خوبی دارند. برای مثال، در برخی از نمونه‌ها که تومور به‌صورت واضح و بزرگ در تصویر قابل مشاهده است، مدل به‌خوبی توانسته ناحیه تومور را تشخیص داده و آن را به‌صورت دقیق شبیه‌سازی کند. این نشان‌دهنده این است که مدل توانسته ویژگی‌های مرتبط با تومور را در تصاویر به‌خوبی بیاموزد.

اما با این حال، در برخی از نمونه‌ها مشکلاتی نیز مشاهده شد. در چند نمونه، ماسک‌های پیش‌بینی شده مدل به‌صورت ناقص بودند و نواحی تومور را به‌طور کامل پوشش نداده بودند. این نشان می‌دهد که مدل در تشخیص برخی از تومورها، به‌ویژه تومورهایی که ممکن است شکل‌های نامنظم داشته باشند یا به بافت‌های اطراف بسیار مشابه باشند، کمی دچار مشکل است. همچنین، در برخی از موارد مدل نواحی کوچکی را به اشتباه به‌عنوان تومور شناسایی کرده بود، که این موضوع نیز ناشی از شباهت بالای برخی از بافت‌های نرمال مغزی به نواحی تومور در تصاویر MRI می‌باشد.

در ردیف چهارم، تصاویر خام همراه با ماسک پیش‌بینی شده مدل نشان داده شده‌اند. این تصاویر ترکیبی به ما کمک می‌کنند که بهتر ببینیم که مدل چگونه نواحی تومور را در تصویر اصلی مشخص کرده است. در بیشتر نمونه‌ها، مدل توانسته است با دقت بالایی نواحی تومور را شناسایی کند و این نواحی را به رنگ قرمز مشخص کرده است. تطابق خوب بین ماسک‌های پیش‌بینی شده و ماسک‌های واقعی در بیشتر تصاویر نشان می‌دهد که مدل ما عملکرد مناسبی در شناسایی تومورها داشته است.

پاسخ ۲ – تشخیص تابلو های راهنمایی و رانندگی

۱-۲. آماده سازی مجموعه داده

مجموعه داده **German Traffic Sign Detection Benchmark (GTSDB)** شامل تصاویری از علائم ترافیکی در شرایط واقعی است و برای شناسایی و تشخیص علائم ترافیکی طراحی شده است. این مجموعه برای اولین بار در سال ۲۰۱۳ ارائه شده است.

این مجموعه شامل ۹۰۰ تصویر با فرمت ppm. از صحنه‌های طبیعی ترافیکی است. علائم به ۴۳ کلاس مختلف تقسیم‌بندی شده‌اند که هر کلاس نمایانگر نوع خاصی از علامت ترافیکی است (مانند محدودیت سرعت، علائم خطر، علائم اجباری و غیره). همینطور این 43 کلاس نیز به 4 کلاس کلی تقسیم شده‌اند که قرار است در ادامه از این 4 کلاس استفاده شود.

همچنین این دیتاست شامل فایل gt.txt است که شامل اطلاعات زیر است:

#ImgNo#;#leftCol#;#topRow#;#rightCol#;#bottomRow#;#ClassID#

ImgNo : شماره تصویر.

leftCol, topRow, rightCol, bottomRow : مختصات مستطیل محدوده‌ی علامت.

ClassID : شناسه‌ی عددی نوع علامت.

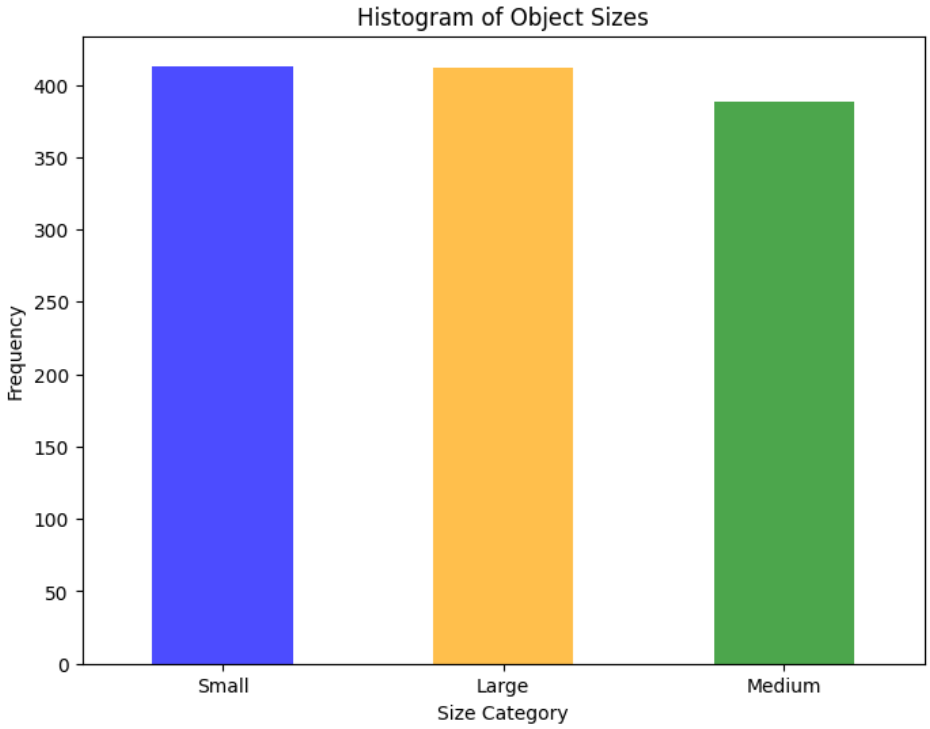
تصاویر اصلی به شکل زیر هستند:

شکل ۹نمونه هایی از تصاویر اصلی

شکل ۱۰نمونه هایی از تصاویر اصلی

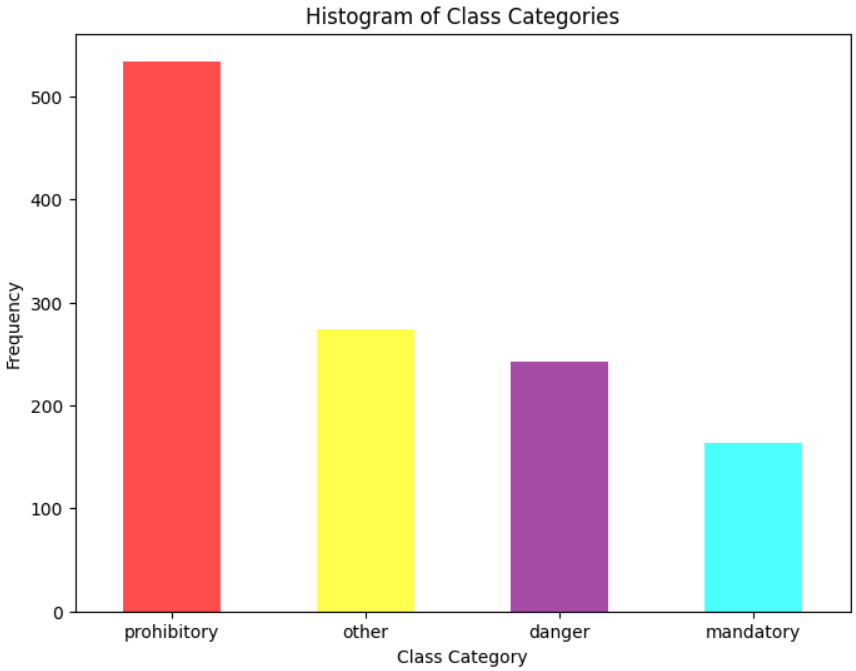
شکل ۱۱ نمونه هایی از تصاویر اصلی

با توجه به آستانه های بیان شده در مقاله، برای سه دسته small، medium و large هیستوگرام‌ها به شکل زیر هستند:



شکل ۱۲. هیستوگرام مربوط به توزیع اندازه اشیا برای کل داده‌ها

دیده می‌شود که فراوانی هر سه دسته تقریبا با هم برابر است و تقریبا imbalance ی وجود ندارد.



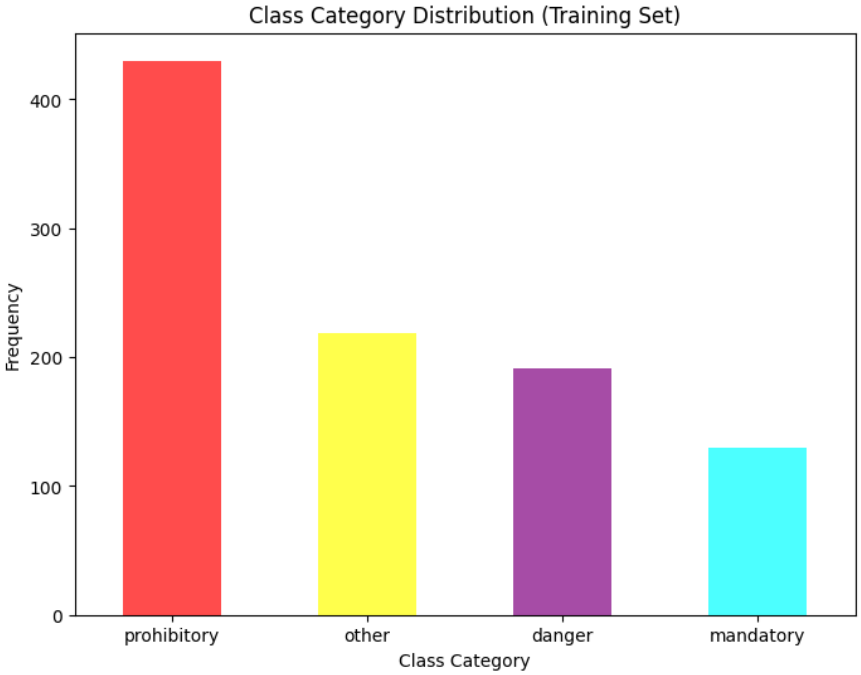
شکل ۱۳. هیستوگرام مربوط به توزیع کلاس ها برای کل داده‌ها

همانطور که دیده می‌شود دسته prohibitory بیشترین فراوانی و دسته mandatory کمترین فراوانی را نسبت به دسته‌ها دارند.

بعد از تقسیم مجموعه داده به داده‌های آموزش و ارزیابی هیستوگرام مربوط به توزیع کلاس ها و اندازه اشیا را برای آن‌ها را رسم کردیم:

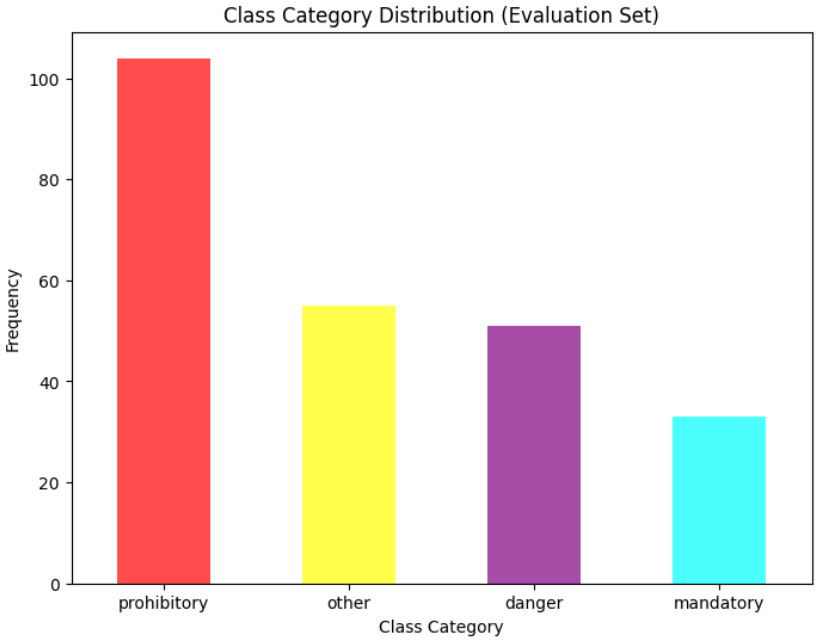
شکل ۱۴. هیستوگرام مربوط به توزیع اندازه اشیا برای داده‌های آموزش

فراوانی برای هر سه دسته برای داده‌های آموزش برابر است. چون هنگام تقسیم، داده‌ها را جوری تقسیم کرده‌ایم که نسبت داده‌ها برای ۴ برچسبی که در داده اولیه بوده‌اند حفظ شود.

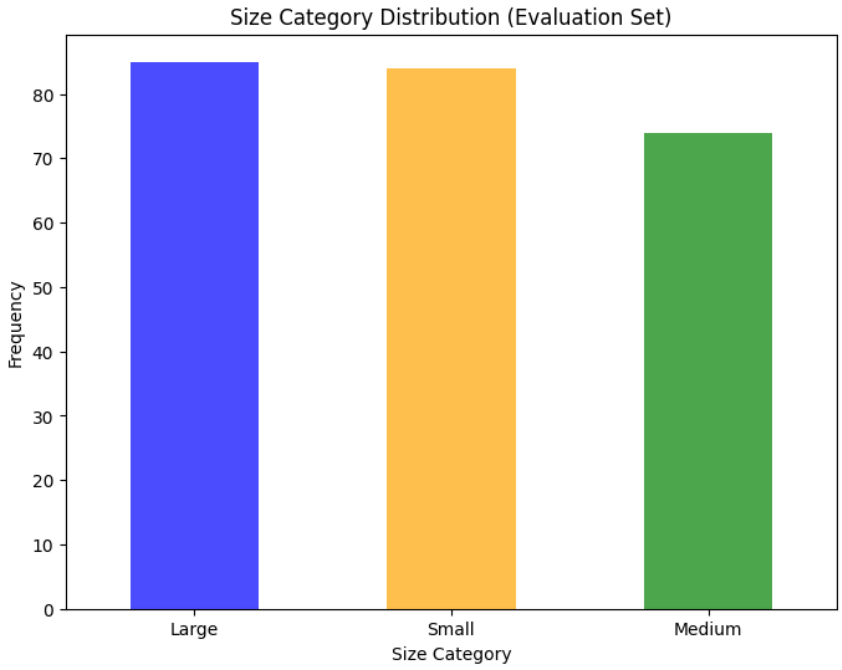


شکل ۱۵. هیستوگرام مربوط به توزیع کلاس ها برای داده‌های آموزش

همانطور که دیده می‌شود نسبت داده‌ها برای ۴ کلاس حفظ شده است.



شکل ۱۶. هیستوگرام مربوط به توزیع کلاس ها برای داده‌های ارزیابی



شکل ۱۷. هیستوگرام مربوط به توزیع اندازه اشیا برای داده‌های ارزیابی

۲-۲. تنظیم دقیق و ارزیابی مدل تشخیص شی دو مرحله ای

❖ در ابتدا توضیح مختصری در مورد این مدل و شبکه پشتیبان آن مطرح کنید.

مدل **Faster R-CNN** یکی از پیشرفته‌ترین و پرکاربردترین مدل‌ها برای شناسایی اشیا **(Object Detection)** است. این مدل یک معماری **دو مرحله‌ای** دارد:

۱. **مرحله اول :** یک شبکه پیشنهاد منطقه **(RPN)** برای پیش‌بینی مناطق مستطیلی **(Region Proposals)** که احتمال دارد اشیایی در آن‌ها وجود داشته باشد.

۲. **مرحله دوم :** این مناطق توسط یک شبکه **CNN** پردازش می‌شوند تا کلاس شی و مکان دقیق آن مشخص شود.

شبکه پشتیبان **(Backbone)** در اینجا **ResNet50-FPN** است که نقش استخراج ویژگی‌ها از تصویر را بر عهده دارد.

- **ResNet50** : یک شبکه عمیق شامل ۵۰ لایه است که با استفاده از مفهوم **مسیرهای باقی‌مانده (Residual Connections)** باعث می‌شود یادگیری در لایه‌های عمیق‌تر راحت‌تر صورت بگیرد.
- **FPN (Feature Pyramid Network)** : رویکردی برای ترکیب اطلاعات ویژگی در سطوح مختلف (از جزئیات ریز تا کلیات تصویر) است. این کار کمک می‌کند تا مدل بتواند اشیایی با اندازه‌های مختلف را بهتر شناسایی کند.

ترکیب **ResNet50** با **FPN** باعث می‌شود ویژگی‌های غنی و چندمقیاسی برای ورودی به **RPN** و شبکه تشخیص فراهم شود، که در نهایت دقت شناسایی اشیا در تصاویر پیچیده بالاتر می‌رود.

❖ حال اقدامات لازم جهت آماده سازی مدل برای تنظیم دقیق را انجام دهید و به طور کامل این اقدامات شرح دهید.

۱. **بارگذاری مدل از پیش‌آموزش‌دیده**

در این خط، مدل **Faster R-CNN** با شبکه پشتیبان **ResNet50-FPN** که روی دیتاست COCO از پیش آموزش دیده، بارگذاری شده است. این مدل می‌تواند به عنوان نقطه شروع برای تنظیم دقیق روی یک مجموعه داده جدید استفاده شود.

تغییر سر شبکه(Head)

۲. **تغییر سر شبکه (Head)**

تعداد کلاس‌ها در مجموعه داده را در متغیر مشخص می‌کنیم که در ادامه استفاده کنیم، که شامل ۴ کلاس و کلاس پس‌زمینه (background) می‌شود. و در مجموع می‌شود ۵ کلاس. که خود مدل FasterRCNN برچسب 0 را برای پس‌زمینه نگه می‌دارد.

سپس تعداد ویژگی‌های ورودی که Head نیاز دارد، از مدل اصلی بازیابی می‌شود.

بعد با استفاده از **FastRCNNPredictor** ، Head قدیمی حذف و یک Head جدید با تعداد 5 کلاس ساخته می‌شود. این کار باعث می‌شود مدل بتواند کلاس‌های جدید را پیش‌بینی کند.

❖ داده‌های تقسیم‌بندی شده برای آموزش مدل را آماده‌سازی کنید، و عملیات نرمالسازی را روی داده‌ها انجام دهید.

در این بخش یک کلاس سفارشی نوشتم به نام GTSDBDataset که داده‌ها را از یک مجموعه فایل‌های تصویری و اطلاعات مربوط به bounding boxes (موجود در فایل annotations) را بارگذاری می‌کند.

سپس عملیات و نرمالسازی‌هایی که بر روی تصاویر انجام دادم تبدیل تصاویر به فرمت Tensor با محدوده مقداری [0, 1] بود. نرمالسازی داده را انجام ندادم به علت اینکه داخل خود مدل نرمالساز تصویر و همینطور تبدیل کننده سایز تصویر وجود داشت. البته تصاویر ما از نظر اندازه تقریبا در همان محدوده تصاویر ورودی این مدل هستند.

سپس DataLoader ها را نوشتم که در آن Batch Size را برابر ۴ قرار دادم. سپس collate_fn را تعریف کردم تا ورودی‌هایی با ابعاد متغیر (تعداد جعبه‌ها در هر تصویر) به درستی در Batch سازمان‌دهی شوند.

❖ به منظور ارزیابی مدل معیارهای IoU و mAP را پیاده‌سازی کرده و آن ها را شرح دهید.

- IoU (Intersection over Union)

معیار IoU میزان همپوشانی بین یک جعبه پیش‌بینی‌شده (Predicted Box) و جعبه مرجع یا واقعی (Ground Truth Box) را اندازه‌گیری می‌کند. به‌عبارت ساده، نسبت مساحت همپوشانی (Intersection) به مساحت کل (Union) دو جعبه است.

هنگام نوشتن کد اختلاف عرض و ارتفاع را به علاوه ۱ کردیم که محاسبه دقیق‌تر شود و پیکسل های ابتدایی و انتهایی در بازه نیز محاسبه شوند.

فرمول آن به شکل زیر است:

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

- $IoU \in [0,1]$
- $IoU=1$: پیش‌بینی و جعبه واقعی کاملاً همپوشان هستند.
- $IoU=0$: هیچ همپوشانی‌ای بین دو جعبه وجود ندارد.

و به این شکل استفاده می‌شود که معمولاً آستانه‌ای مثل 0.5 یا 0.75 انتخاب می‌شود. اگر IoU پیش‌بینی‌شده بالای این مقدار باشد، به عنوان پیش‌بینی درست (True Positive) در نظر گرفته می‌شود.

- mAP (Mean Average Precision)

mAP نشان‌دهنده میانگین دقت پیش‌بینی مدل برای تمامی کلاس‌ها در آستانه‌های مختلفIoU است. در واقع از میانگین گیری AP همه کلاس‌ها به دست می‌آید.

Average Precision (AP) نماینده دقت مدل در سطوح مختلف Recall است و به‌صورت میانگین وزنی محاسبه می‌شود. به طور خلاصه AP می‌شود مساحت زیر منحنی Precision-Recall که Precision در محور عمودی و Recall در محور افقی قرار دارد.

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N}$$

که N تعداد کلاس‌ها است.

معیار mAP از نظر مفهومی ترکیبی از کیفیت موقعیت جعبه‌ها و شناسایی صحیح کلاس‌ها است. یعنی مدل‌هایی با mAP بالا، هم جعبه‌ها را دقیق‌تر پیش‌بینی می‌کنند و هم کلاس‌ها را به‌درستی شناسایی می‌کنند.

- ❖ حال با استفاده از بهینه ساز و تابع هزینه مناسب (دلیل انتخاب خود را توضیح دهید) اقدام به تنظیم دقیق مدل کنید.

برای آموزش این مدل از بهینه‌ساز SGD (Stochastic Gradient Descent) استفاده کرده‌ام. SGD وزن‌ها را به کمک گرادیان نزولی و با استفاده از نمونه‌های تصادفی (Batch) ها به‌روزرسانی می‌کند . SGD برای تنظیم دقیق (Fine Tune) مدل‌هایی که از قبل آموزش‌دیده‌اند، عملکرد بهتری نشان می‌دهد. این روش نسبت به بهینه‌سازهایی مثل Adam احتمال کمتری دارد که در داده‌های جدید بیش از حد Overfitting پیدا کند و هم سرعت همگرایی و هم توانایی عمومی‌سازی را بهبود می‌بخشد.

برای آموزش مدل از نرخ یادگیری 0.005 استفاده شده است.

Faster R-CNN به طور پیش‌فرض از ترکیبی از چندین تابع هزینه استفاده می‌کند. در واقع این توابع داخل مدلی که بارگذاری کرده‌ایم وجود دارند.

Classification Loss:

تابع Cross-Entropy Loss برای پیش‌بینی کلاس اشیا استفاده شده است و کمک می‌کند تا مدل کلاس هر جعبه را به درستی پیش‌بینی کند.

Regression Loss:

یک Smooth L1 Loss برای اصلاح مختصات جعبه‌های محدودکننده (Bounding Boxes) استفاده شده است این تابع به مدل یاد می‌دهد که جعبه پیش‌بینی‌شده را با جعبه واقعی هم‌تراز کند. و به دلیل حساسیت کمتر به‌Outlier ها نسبت به‌L2 Loss ، انتخاب خوبی برای تنظیم دقیق جعبه‌هاست.

که در نهایت هنگام آموزش همه خطاها را با هم جمع می‌کنیم.

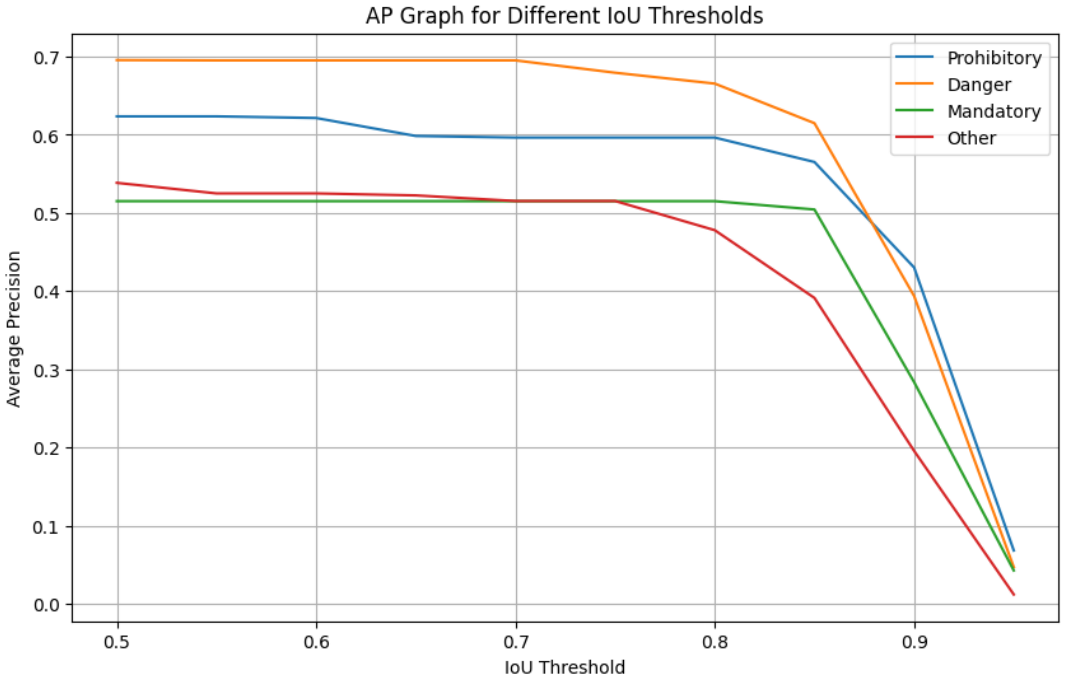
و در آخر هم مدل را با قرار دادن یک کنترل کننده برای آموزش به اندازه 10 ایپاک آموزش دادم.

❖ پس از تنظیم دقیق مدل، اقدام به ارزیابی آن با استفاده از معیارهای پیاده‌سازی شده کنید. (IoU = 0.5)

مقدار به دست آمده برای این معیار و برای این مدل بر روی داده‌های ارزیابی برابر 0.5932 شده است که مقدار نسبتا خوبی است. و نشان می‌دهد که مدل تعداد نسبتا خوبی از جعبه‌ها را توانسته است که تشخیص دهد.

❖ نمودار مربوط به AP به ازای IoU های متفاوت را مشابه Figure 1 مقاله برای هرکلاس رسم کنید و تحلیل کنید. (نمودار باید شامل همه کلاس ها باشد!)

نمودار ترسیم شده برای AP کلاس‌ها به شکل زیر است و به ازای IoU های از 0.5 تا 0.95 رسم شده است.



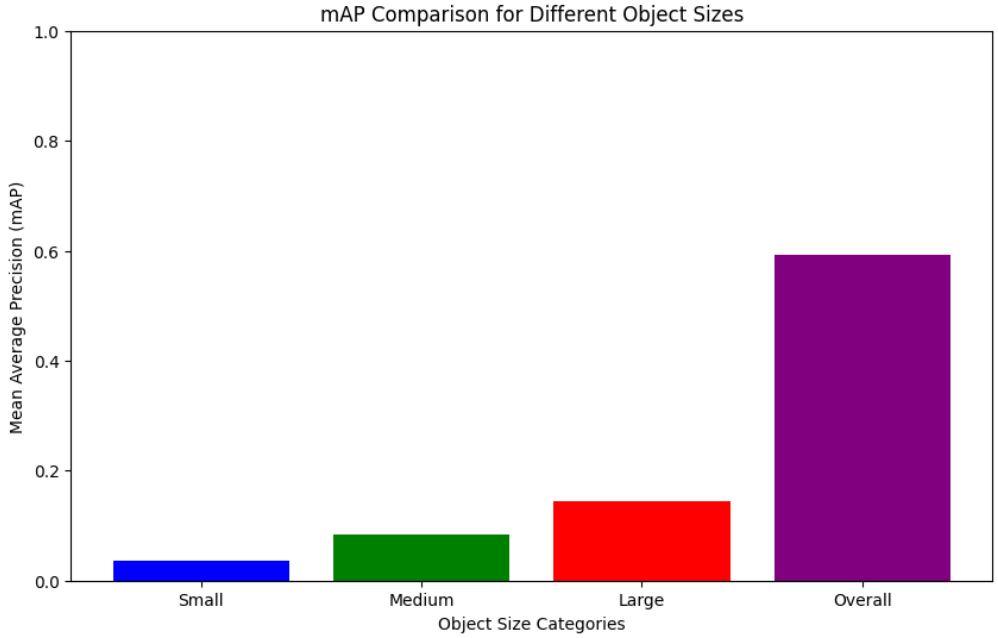
شکل ۱۸. نمودار مربوط به **AP** به ازای **IoU** های متفاوت برای مدل **FasterRCNN**

طبق نمودار بالا، بهترین پیش‌بینی‌های مدل برای دسته Prohibitory بوده است و ضعیف‌ترین پیش‌بینی در بیشتر مقادیر IoU برای دسته Other بوده است. و همه آن‌ها با زیاد شدن threshold میزان AP آن‌ها رفته رفته کم شده است تا نزدیک به صفر شود که یعنی نمی‌توان از مدل انتظار تطبیق کامل را

داشته و این مقدار کم شدن هم از threshold تقریبا 0.85 شدت گرفته است و تا قبل از آن برای تقریبا تمام کلاس‌ها تغییرات IoU تاثیرات خاصی در AP کلاس‌ها نداشته است.

❖ اقدام به ارزیابی مدل برای اشیا با اندازه‌های متفاوت کنید و نموداری مشابه **Figure 2** را برای آن ترسیم کنید و نتایج را تحلیل کنید.

مدل را برای سه دسته از اندازه‌ها به صورت جداگانه ارزیابی کردیم که نمودار زیر حاصل شد.



این نمودار مشخص است که مقدار mAP کلی خیلی زیاد است و به طور متوسط تشخیص مدل خوب است. ولی برای اندازه‌های متفاوت این مقادیر کم و متفاوت می‌شود. همانطور که دیده می‌شود مدل تابلوهای بزرگتر را بهتر از تابلوهای کوچکتر تشخیص می‌دهد و با کوچکتر شدن ابعاد تابلوها، تشخیص آن‌ها برای مدل سخت‌تر می‌شود و کیفیت تشخیص پایین می‌آید.

❖ یک نمونه تصویر از داده‌های ارزیابی را با استفاده از مدل تنظیم شده، پیش بینی کنید و نتایج را به مانند تصویر زیر نمایش دهید.

چند نمونه از تصاویر تولید شده به شکل زیر هستند:

توضیح: برای احتیاط و تمیز شدن کار، یک threshold ای بر روی میزان score هر جعبه برای هر تصویر در نظر گرفته شده است. البته تصویر آخر برای نشان دادن تصویر خروجی از مدل، بدون threshold است.

شکل ۱۹. نمونه تصویر از داده‌های ارزیابی، با استفاده از مدل FasterRCNN تنظیم شده با اعمال threshold

شکل ۲۰. نمونه تصویر از داده‌های ارزیابی، با استفاده از مدل **FasterRCNN** تنظیم شده با اعمال **threshold**

شکل ۲۱. نمونه تصویر از داده‌های ارزیابی، با استفاده از مدل **FasterRCNN** تنظیم شده بدون اعمال **threshold**

در تصاویر مشاهده می‌شود که مدل خیلی خوب توانسته است جعبه‌ها را پیدا و با احتمال تقریبا 1 کلاس آن‌ها را تشخیص دهد.

۳-۲. تنظیم دقیق و ارزیابی مدل تشخیص شی تک مرحله ای

تمامی مراحل ذکر شده برای قسمت بالا در این بخش انجام شده است و صرفا از بیان بخش‌های تکرار صرف نظر می‌شود، چون یک سری از بخش‌ها دقیقا همان‌هایی است که در بخش قبل انجام شده و نتایج نیز همان است.

❖ در ابتدا توضیح مختصری در مورد این مدل و شبکه پشتیبان آن مطرح کنید.

مدل (Single Shot MultiBox Detector) **SSD300** یک مدل تشخیص اشیا تک‌مرحله‌ای است که برای شناسایی اشیا در تصاویر استفاده می‌شود. این مدل با استفاده از ساختار شبکه **VGG16** به عنوان شبکه پشتیبان(backbone) ، ویژگی‌های تصویر را استخراج کرده و سپس از چندین لایه اضافی برای تشخیص اشیا در مقیاس‌های مختلف استفاده می‌کند.

این مدل با پیش‌بینی جعبه‌های محدودکننده و کلاس‌های اشیا در یک مرحله، سرعت بالایی دارد و برای شناسایی اشیا در مقیاس‌های مختلف از لایه‌های اضافی کانولوشنی استفاده می‌کند SSD300 .قابلیت شناسایی اشیا در اندازه‌های مختلف را به‌طور همزمان دارد و ورودی ۳۰۰ x 300 پیکسل به آن اجازه می‌دهد تا در زمان کوتاه‌تری به پردازش تصویر بپردازد. برخلاف مدل‌های دومرحله‌ای مانندFaster R-CNN ، SSD300 با سرعت بیشتری به پردازش و شناسایی می‌پردازد، هرچند همانطور که خواهیم دید دقت آن به‌ویژه در شناسایی اشیا کوچک ممکن است پایین‌تر باشد.

شبکه پشتیبان VGG16 یک معماری معروف شبکه عصبی کانولوشنی است که شامل ۱۶ لایه (۱۳ لایه کانولوشنی و ۳ لایه کاملاً متصل) است. در مدل‌های تشخیص اشیا مانند SSD300، از VGG16 به عنوان استخراج‌کننده ویژگی‌های تصویر استفاده می‌شود. لایه‌های کانولوشنی این شبکه قادر به استخراج ویژگی‌های سطح پایین و پیچیده از تصاویر هستند که سپس برای شناسایی اشیا در مقیاس‌ها و موقعیت‌های مختلف مورد استفاده قرار می‌گیرند. درSSD300 ، لایه‌های کاملاً متصل VGG16 حذف شده و به جای آن لایه‌های کانولوشنی اضافی برای پیش‌بینی جعبه‌های محدودکننده و کلاس اشیا اضافه می‌شود.

- ❖ حال اقدامات لازم جهت آماده سازی مدل برای تنظیم دقیق را انجام دهید و به طور کامل این اقدامات شرح دهید.

۱. بارگذاری مدل از پیش‌آموزش‌دیده

از آن جایی که مدل مورد نیاز به صورت کامل از پیش به صورت آموزش داده شده وجود داشت، مدل را بارگذاری کردیم. حال یک مدل داریم که ویژگی‌های پایه‌ای تصویر مانند لبه‌ها، الگوها و بافت‌ها را یاد گرفته است.

۲. تنظیم تعداد کلاس‌ها

حال تغییراتی را در Head طبقه‌بندی (Classification Head) مدل ایجاد می‌کنیم. تعداد کلاس‌ها را برای ۵ کلاس تنظیم می‌کنیم (که شامل پس‌زمینه و ۴ کلاس مختلف از کلاس‌های کلی علائم ترافیکی می‌شود).

- ❖ داده‌های تقسیم‌بندی شده برای آموزش مدل را آماده‌سازی کنید، و عملیات نرمالسازی را روی داده‌ها انجام دهید.

در اینجا همان کارهایی که در بالا انجام شد انجام شدند و نرمالسازی خاصی بر روی داده‌ها انجام نشد، چون این مدل نیز داخلش داده‌ها را نرمال می‌کند فقط چون ورودی این مدل ابعاد 300x300 است تصاویر و ابعاد جعبه‌ها را به این مقدار کاهش دادیم.

❖ به منظور ارزیابی مدل معیارهای IoU و mAP را پیاده‌سازی کرده و آن ها را شرح دهید.

پیاده‌سازی و تعریف آن در بخش قبلی انجام شد.

❖ حال با استفاده از بهینه ساز و تابع هزینه مناسب (دلیل انتخاب خود را توضیح دهید) اقدام به تنظیم دقیق مدل کنید.

از همان بهینه‌سازی و تابع هزینه‌ای استفاده شد که در بخش قبل گفته شده بود. فقط از نرخ یادگیری 0.0004 برای آموزش مدل استفاده شده است. و مدل در 10 آموزش داده شده است.

و برای توابع هزینه هم از همان‌هایی که در بالا بیان شد و به همان صورت استفاده شد.

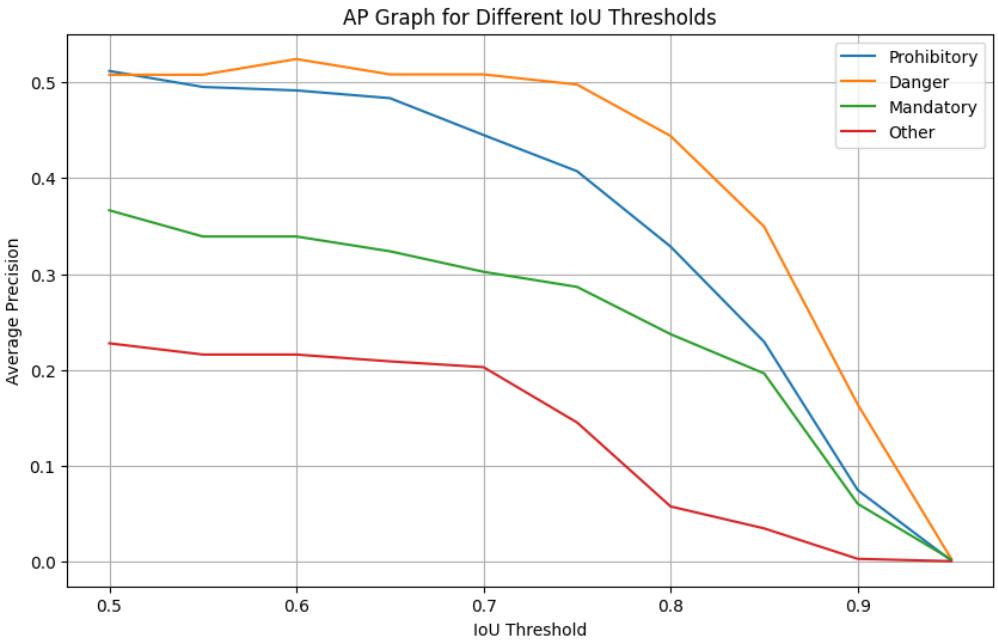
❖ پس از تنظیم دقیق مدل، اقدام به ارزیابی آن با استفاده از معیارهای پیاده‌سازی شده کنید. (IoU = 0.5)

مقدار به دست آمده برای این معیار و برای این مدل بر روی داده‌های ارزیابی برابر 0.4035 شده است که مقدار نسبتا بدی است. و نشان می‌دهد که مدل دقت خوبی برای تشخیص جعبه‌ها ندارد.

البته برای دقت معقول‌تر و همچنین جلوگیری از شلوغ شدن بی‌خودی تصاویر جعبه‌هایی که امتیاز اطمینان کمتر از 0.1 داشتند حذف شدند. چون این مدل هم دقت کمتری دارد و هم تعداد جعبه‌های زیادی را تولید می‌کند که احتمال خیلی خیلی کوچکی هم دارند.

❖ نمودار مربوط بهAP به ازایIoU های متفاوت را مشابهFigure 1 مقاله برای هرکلاس رسم کنید و تحلیل کنید. (نمودار باید شامل همه کلاس ها باشد!)

نمودار ترسیم شده برای AP کلاس‌ها به شکل زیر است و به ازای IoU های از 0.5 تا 0.95 رسم شده است.



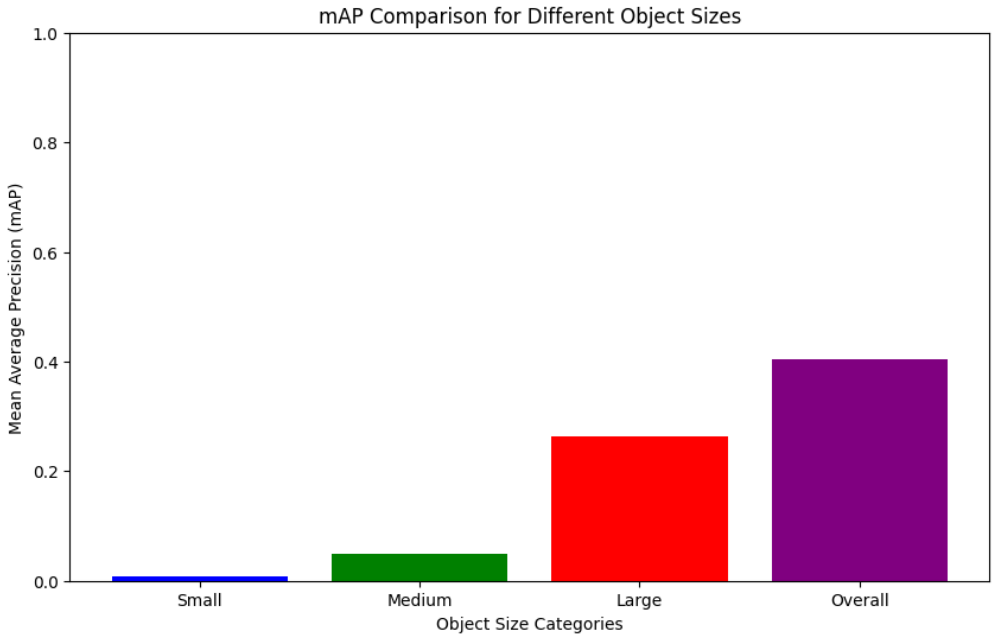
شکل ۲۲. نمودار مربوط به **AP** به ازای **IoU** های متفاوت برای مدل **SSD300**

طبق نمودار بالا، بهترین پیش‌بینی‌های مدل برای دسته Prohibitory بوده است و ضعیف‌ترین پیش‌بینی در بیشتر مقادیر IoU برای دسته Other بوده است. و همه آن‌ها با زیاد شدن threshold میزان AP آن‌ها رفته رفته کم شده است تا نزدیک به صفر شود که یعنی نمی‌توان از مدل انتظار تطبیق کامل را داشته و این مقدار کم شدن هم از threshold های مختلف برای هر کلاس شدت گرفته است و برای هر کلاس متفاوت است. تفاوت این معیار و فاصله نمودارها در این شکل بیشتر از مدل قبل است.

در دو مدل می‌توان کلاس other را پایین‌تر از همه و دید و کلاس Prohibitory را بالاتر از همه که یکی از دلایل اصلی آن می‌تواند زیاد بودن دسته Prohibitory و نسبتا کم و متنوع‌تر بودن دسته Other باشد.

در این تصویر می‌توان مشاهده کرد که نمودارها سریع تر و با threshold های پایین‌تر شروع به کم شدن برای AP کرده‌اند.

❖ اقدام به ارزیابی مدل برای اشیا با اندازه‌های متفاوت کنید و نموداری مشابه Figure 2 را برای آن ترسیم کنید و نتایج را تحلیل کنید.



در این مدل نیز دیده می‌شود که مقدار mAP کلی خیلی زیاد نیست و به طور متوسط تشخیص مدل خیلی خوب نیست. و برای اندازه‌های متفاوت این مقادیر کم و متفاوت می‌شود. همانطور که دیده می‌شود مدل تابلوهای بزرگتر را بهتر از تابلوهای کوچکتر تشخیص می‌دهد و با کوچکتر شدن ابعاد تابلوها، تشخیص آن‌ها برای مدل سخت‌تر می‌شود و کیفیت تشخیص پایین می‌آید. حتی در اینجا اختلاف بدتر شدن تشخیص برای ابعاد به غیر از بزرگ خیلی بیشتر هم هست. یعنی این مدل تصاویر با ابعاد کوچک را خیلی بدتر تشخیص می‌دهد.

❖ یک نمونه تصویر از داده‌های ارزیابی را با استفاده از مدل تنظیم شده، پیش بینی کنید و نتایج را به مانند تصویر زیر نمایش دهید.

چند نمونه از تصاویر تولید شده به شکل زیر هستند:

توضیح: برای احتیاط و تمیز شدن کار، یک threshold ای بر روی میزان score هر جعبه برای هر تصویر در نظر گرفته شده است. البته تصویر آخر برای نشان دادن تصویر خروجی از مدل، بدون threshold است.

همچنین تصاویر بعد از خروجی گرفتن از مدل، دوباره به اندازه اولیه خود بازگردانده شده‌اند.

شکل ۲۳. نمونه تصویر از داده‌های ارزیابی، با استفاده از مدل **SSD300** تنظیم شده با اعمال **threshold**

شکل ۲۴. نمونه تصویر از داده‌های ارزیابی، با استفاده از مدل **SSD300** تنظیم شده با اعمال **threshold**

شکل ۲۵. نمونه تصویر از داده‌های ارزیابی، با استفاده از مدل **SSD300** تنظیم شده بدون اعمال **threshold** هنگام رسم تصویر

در تصاویر مشاهده می‌شود که به نظر مدل خیلی خوب توانسته است جعبه‌ها را پیدا و اما احتمال تشخیص کلاس‌ها در آن پایین است، و حتی در بعضی جاها کلاس را اشتباهی تشخیص داده است.

۴-۲. ارزیابی نتایج و مقایسه مدل ها

- **کدام مدل عملکرد بهتری در شناسایی علائم کلاس‌های مختلف داشته است؟ تحلیل خود را ارائه دهید.**

مدل FasterRCNN دقت بهتری در شناسایی علائم کلاس‌های مختلف داشته است. و دلیل آن هم می‌توان باتوجه به نتایج به دست آمده از ارزیابی‌ها فهمید. مثلا مدل FasterRCNN میزان mAP خیلی بیشتری داشت که در آن کیفیت تشخیص کلاس‌ها نیز لحاظ شده است و یا در نمودار AP نیز می‌توان دید که مقدار کلاس‌ها هر کدام میزان AP بیشتری داشتند و در thresholdهای بالاتر هم این مقدار را تا حد خوبی حفظ کرده بودند ولی در مدل SSD300 این گونه نبود.

همینطور کیفیت مدل FasterRCNN را می‌توان از نمونه عکس‌های تولیدی به نسبت عکس‌های تولیدی مدل SSD300 مشاهده کرد.

- **کدام مدل عملکرد بهتری در شناسایی علائم با اندازه‌های مختلف داشته است؟ تحلیل خود را ارائه دهید.**

مدل FasterRCNN باتوجه به نمودار mAP که برای تشخیص تصاویر با سه اندازه مختلف رسم شده بود و مقادیری که برای هر کدام به دست آمده بود(مقادیر مربوط به مدل FasterRCNN بیشتر بود)، عملکرد بهتری داشته است.

- **به نظر شما چه عواملی می‌تواند بر عملکرد مدل‌ها تأثیر گذاشته باشد؟ راهکارهای پیشنهادی خود را برای بهبود دقت و سرعت مدل‌ها ارائه دهید.**

یکی از عوامل تاثیرگذار معماری خود مدل‌ها است، همانطور که در بالا دیده شد، مدل با معماری دو مرحله‌ای عملکرد بهتری از نظر دقت داشت و قابلیت تنظیم دقیق بهتری نیز داشت. ولی سرعت آموزش برای مدل FasterRCNN که مدل اول بود، کمتر بود.

مورد دیگر که می‌تواند تاثیرگذار باشد معماری backbone ای است که برای آن‌ها استفاده می‌شود. همانطورکه قبل‌تر نیز توضیح داده شد یک سری ازbackbone ها اطلاعات بیشتر و بهتری از تصاویر استخراج می‌کنند و حتی می‌توانند برای تصاویر ریزتر هم عملکرد خوبی داشته باشند. که می‌توان از backbone هایی که کوچک هستند ولی ویژگی‌های خیلی خوبی را استخراج می‌کنند استفاده کرد که هم سرعت و هم دقت مدل را بیشتر کرد.

در اینجا داده‌ها برای انجام تنظیم دقیق خیلی کم نبود ولی در میزان کیفیت پایین تاثیر داشت. در واقع تعداد داده‌های بیشتر می‌تواند دقت مدل‌ها را بهبود بخشد که البته با همین تعداد داده نیز می‌توان از augmentation استفاده کرد و تعداد داده‌ها را بیشتر کرد.

مورد دیگری که می‌تواند بر روی بهبود دقت و سرعت آموزش تاثیرگذار باشد، تنظیم درست و بهینه هایپرپارامترها است.

همینطور می‌توان از یک سری روش‌ها برای بهبود روند آموزش و جلوگیری از از بین رفتن گرادیان و یا یک سری regularization ها استفاده کرد.